SPACE LAB SYSTEM ANALYSIS

INTERIM FINAL REPORT
SEPTEMBER 1986 - SEPTEMBER 1987

(NASA-CR-179207) SPACE LAB SYSTEM ANALYSIS N88-10869
Interim Final Report, Sep. 1986 - Sep. 1987
(Mississippi State Univ.) 192 p Avail:
NTIS HC A09/MF A01 CSCL 22B Unclas
G3/18 0106043

Submitted by:

F. M. Ingels, Principal Investigator
T. B. Rives, Associate Investigator

Mississippi State University
Electrical Engineering Department
Mississippi State, MS 39762
(601) 325-3912

Submitted to:

NASA/MSFC-EB32/JOBE
NAS8-36717
(205) 544-3555

MISSISSIPPI STATE UNIVERSITY
COLLEGE OF ENGINEERING

DEPARTMENT OF ELECTRICAL ENGINEERING
DRAWER EE
MISSISSIPPI STATE, MISSISSIPPI 39762
PHONE (601) 325-3912

October 1, 1987

NASA
AS24-D
Marshall Space Flight Center, AL 35812

Dear Sir:

With regards the NASA Contract NAS8-36717, enclosed is the final technical report for the first phase of this contract.

If you have any questions, please contact either of the signatories.

Sincerely,

Frank Ingels, Ph.D.
Principal Investigator

Teresa B. Rives
Associate Investigator

FI:asr

CODE:

AP29-H*	0
AS24-D	5
AT01	1
CC01/Wofford	1
EM12A-37/Schrimsher	1
ONRRR*	0
EB32/Jobe	5 + repro.
NASA SCIENTIFIC AND TECHNICAL INFORMATION FACILITY	1 + repro.

* Copy of Letter of Transmittal

NASA
National Aeronautics and
Space Administration

Report Documentation Page

1. Report No.

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
Spacelab System Analysis

5. Report Date
September 1987

6. Performing Organization Code

7. Author(s)
F. M. Ingels and T. B. Rives

8. Performing Organization Report No.
MSU-EE-FIN-9-87

10. Work Unit No.

9. Performing Organization Name and Address
Department of Electrical Engineering
Mississippi State University
Mississippi State, MS 39762

11. Contract or Grant No.
NAS8-36717

13. Type of Report and Period Covered
Interim Final Report
Sept 86 to Sept 87 .

12. Sponsoring Agency Name and Address
National Aeronautics and Space Administration
Washington, DC 20516

14. Sponsoring Agency Code

15. Supplementary Notes
Prepared by Mississippi State University for Sherman Jobe, Huntsville Operation Support Center, Huntsville, Alabama

16. Abstract

An analytical analysis of the HOSC Generic Peripheral processing system has been conducted. This analysis which is Task 1 of the work statement has been fully documented in the October 1986 report and is presented in Section 3.0 of this report. The results are summarized in Table 3.1, Table 3.2, and Table 3.3 and indicate that the maximum delay in performing screen change requests should be less than 2.5 seconds, occurring for a slow VAX host to video screen I/O rate of 50 KBps (Bps = Bytes per second). This delay is due to the average I/O rate of the host computers, the ETHERNET protocol, and the I/O rate from the video terminals to their host computer. Software structure of the main computers and the host computers will have greater impact on screen change or refresh response times.

The HOSC data system model has been updated by a newly coded PASCAL based simulation program which has been installed on the HOSC VAX system. This model is described and documented in Sections 1.0, 2.0, 4.0, 6.0, and the appendices of this report. This model is Task 2 of the work statement.

The October 1986 report and Section 5.0 of this report offer suggestions to fine tune the performance of the ETHERNET interconnection network, Task 3 of the work statement.

Suggestions for using the Nutcracker by Excelan to trace itinerate packets which appear on the network from time to time have been offered in discussions with the HOSC personnel, Task 4 of the statement of work.

Several visits to the HOSC facility were made during the course of the contract to install and demonstrate the simulation model.

17. Key Words (Suggested by Author(s))
ETHERNET
LAN - Local Area Network

18. Distribution Statement

19. Security Classif. (of this report)
Unclassified

20. Security Classif. (of this page)
Unclassified

21. No. of pages
182

22. Price

SPACE LAB SYSTEM ANALYSIS

INTERIM FINAL REPORT
SEPTEMBER 1986 - SEPTEMBER 1987

Submitted by:

F. M. Ingels, Principal Investigator
T. B. Rives, Associate Investigator

Mississippi State University
Electrical Engineering Department
Mississippi State, MS 39762
(601) 325-3912

Submitted to:

NASA/MSFC-EB32/JOBE
NAS8-36717
(205) 544-3555

OVERVIEW OF REPORT

An analytical analysis of the HOSC Generic Peripheral processing system has been conducted. This analysis which is Task 1 of the work statement has been fully documented in the October 1986 report and is presented in Section 3.0 of this report. The results, which are summarized in Table 3.1, Table 3.2, and Table 3.3, indicate that the maximum delay in performing screen change requests should be less than 2.5 seconds, occurring for a slow VAX host to video screen I/O rate of 50 KBps (Bps = Bytes per second). This delay is due to the average I/O rate of the host computers, the ETHERNET protocol, and the I/O rate from the video terminals to their host computer. Software structure of the main computers and the host computers will have greater impact on screen change or refresh response times.

The HOSC data system model has been updated by a newly-coded PASCAL-based simulation program which has been installed on the HOSC VAX system. This model is described and documented in Sections 1.0, 2.0, 4.0, 6.0 and in the appendices of this report. This model is Task 2 of the work statement.

The October 1986 report and Section 5.0 of this report offer suggestions to fine tune the performance of the ETHERNET interconnection network, Task 3 of the work statement.

Suggestions for using the Nutcracker by Excelan to trace itinerate packets which appear on the network from time to time have been offered in discussions with the HOSC personnel, Task 4 of the statement of work.

Several visits to the HOSC facility were made during the course of the contract to install and demonstrate the simulation model.

TABLE OF CONTENTS

APPENDICES

LIST OF TABLES

LIST OF FIGURES

1.0 INTRODUCTION TO LOCAL AREA NETWORKS

In a large computer data base system, many users have access to a data base at any given time. In order to give access to each user, Data Communication Systems such as NASA´s Huntsville Operation Support Center (HOSC) are established. The HOSC system contains many interconnecting networks including a Video Display Interconnection Network which is connected via an ETHERNET Networking System. The system configuration for the Video Display Interconnection Network in different modes of operation will vary depending on the application: operator training for malfunctions, astronaut simulators, video display operations training, launch system monitor engineers, and various displays for data such as sensor data.

Within high volume data systems such as HOSC, data flow bottlenecks occasionally occur with the result that the system is no longer able to handle the information transfer. Faced with this problem, a systems operator, using any analysis tools available, would have to investigate the configuration and make changes to resolve data flow bottlenecks. However, this is not a simple task since minimal changes could require hardware re-cabling, software considerations, and documentation updates. Furthermore, an overall analysis of any proposed system changes would have to be made to determine the effect on launch and mission data bases, the effect of adding or removing system terminals, changes in system throughput, and other aspects affected by the system reconfiguration. The system operator must be able to reasonably predict the system response when reconfigured to avoid creating additional data flow bottlenecks and to insure overall system integrity. This work concerns itself with the analysis and simulation of the ETHERNET Local Area Network (LAN) used for intracomputer data communications in the Video Display Interconnection Network.

The following list defines the characteristics of such a Local Area Network [FKP85]:

1) High data rates (typically 1 to 10 Mbps)
2) Limited geographical scope - typically spanning about kilometer
3) Support of full connectivity - all devices should have the potential to communicate with each other
4) Equal access by all users
5) Ease of reconfiguration and maintenance
6) Good reliability and error characteristics
7) Stability under high loads
8) Compatibility to the greatest possible for various equipment
9) Relatively low cost.

1.1 System Configuration and Operation.

Figure 1.0 presents a generic picture of the system configuration modeled. The main frames have external data sources from which data is received. The information is formatted and packaged for periodic transmission to the Video Terminals. This action will be referred to as a ´screen refresh´ operation. Additionally, large blocks of data are transferred from the main frames to the hosts for automatic storage and backup purposes. The process is called a ´tape file dump.´ The operator of a video terminal randomly performs ´operator requests´ for the system to update his terminal or change his data base. These are the primary activities assumed to be occurring within the system at any point in time.

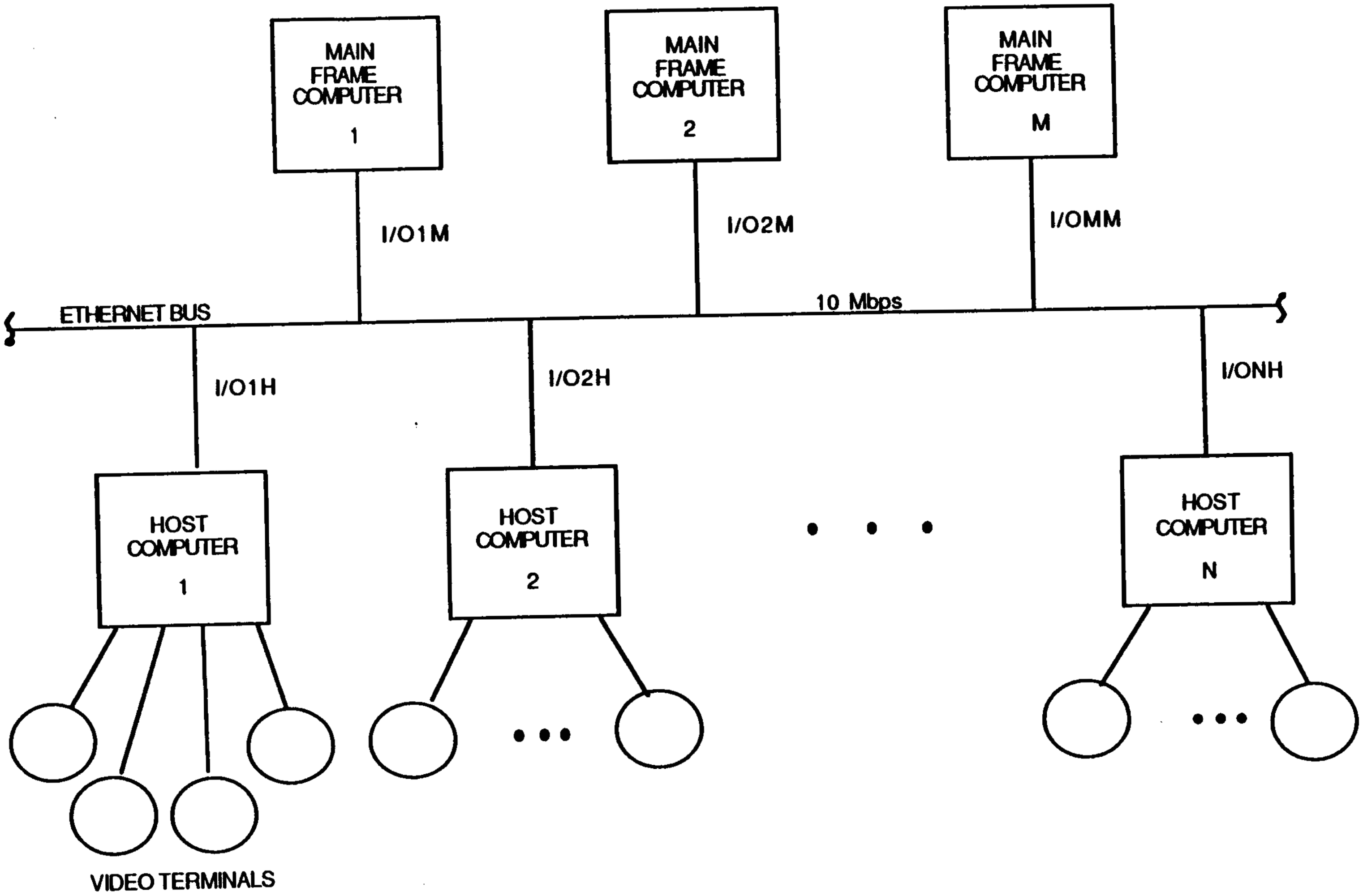

FIGURE 1.0 GENERIC SYSTEM CONFIGURATION

The HOSC configuration is displayed in Figure 1.1 with specific devices included. The external data influx occurs from two sources: a 50 MBps composite date stream through the HRDS and a 4MBps composite data stream through the MDM. These devices are connected to a Data Distribution System which distributes designated data to the VAX 11/780 and to the PE 3254. These two main frames primarily format data, perform data calculations to compute new parameters, perform data unit conversions, and extract data trends. The gathered data is formatted and packaged for presentation to the Video Terminals. This information is distributed via the Ethernet Network to the VAX 11/725 and the VAX 11/730 computers.

The PE 3254 provides the primary computational power for data calculations, whereas the VAX 11/780 provides the main computational power for formatting data to be displayed.

1.2 Limiting Factors.

There are many factors which affect the operation and analysis of a system. The Ethernet effectiveness is affected by the following: the data I/O rates of devices on the system, the memory buffer availability on devices, and the amount of information to be transferred on the bus.

1.2.1 Data Influx.

The aggregate data flow of information from the Data Distribution

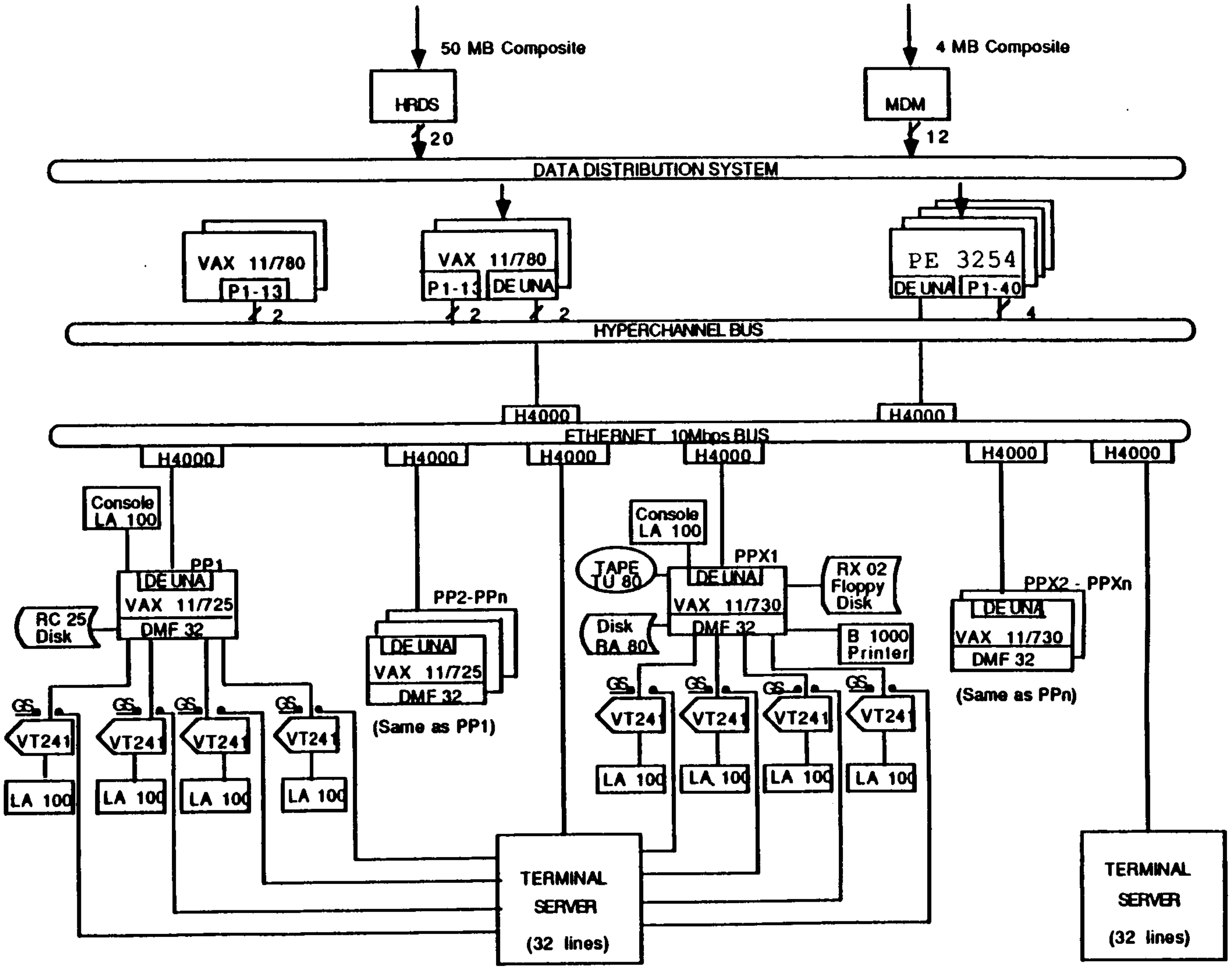

FIGURE 1.1 HOSC PERIPHERAL PROCESSING SYSTEM

System to the two main frames is 2Mbps, or 250kbps i.e. bits per second. Additional data may be gathered by the two computers from the Hyperchannel bus, but the effective influx of data remains at 2Mbps. It is not necessary for all of this data to be scaled, analyzed, and formatted for the Video Display Terminals, but the capability must exist. The quantity of data transmitted on the Ethernet is also affected by formatting and packaging for reception by the Video Terminals. This could increase the bus traffic volume two to three times the raw data influx.

1.2.2 Device Data I/O Rates.

The input/output capabilities of the main frames also affect the system activity. For the VAX 11/780 there are four internal UNIBUS channels each having an effective I/O rate of 1.5MBps (Bytes per second). However, only one channel is assigned for the Space Telescope and Video Terminal communications. Thus, the effective data transfer rate for this I/O function is estimated to be an average of .3MBps due to the multifunction requirements of the bus. Even though the UNIBUS data rate is 1.5MBps and the actual data is transferred at this rate, the slower throughput rate of 300KBps is a more accurate measure of the average achievable I/O rate due to the waiting time when gaining access to the bus.

The PE 3254 has four 10MBps buses with one assigned to Space Telescope and Video Terminal communication. The equivalent average data rate in this case is estimated to be 3MBps since the bus rate is degraded while the bus is performing other required functions. However, when accessing the Ethernet, the average I/O rate is limited to 1.25MBps (including overhead information). Therefore, the effective average data rate is about 1.0MBps. The restriction to 1.25MBps is placed on the system by the Ethernet I/O rate of 1.25MBps or 10Mbps.

The VAX 11/730 has only one 1.5MBps UNIBUS which is required to support all input and output transfers and disk and DMA accesses. The effective communication rate to the Ethernet for this device is estimated to be an average of .3MBps due to the multifunction requirements on the bus.

1.2.3 Device Memory Buffers.

The data rate from the host devices to the individual terminals will not be a factor in the analysis of the Ethernet for several reasons. The VAX host computers must have a considerable amount of buffer storage space in which to place incoming data from the Ethernet since the host accepts data from the Ethernet at a much higher rate than that of the Video Terminals. Therefore, the data must be buffered and sent to the terminal at the terminal I/O rate which is assumed to be much lower than that of the hosts. In addition, the terminal may not be ready to accept a terminal update which may cause the host to backlog the information until a later point. In this case, the operator may not wish the information changed or possibly the operator was performing an operator request thus precluding a screen refresh operation from the main frame. For these reasons, the host computers must have considerable backlog capability. Thus, the host to terminal data transfer does not affect the Ethernet analysis. Table 1.0 summarizes the computer I/O characteristics used in this research.

Table 1.0 I/O Rates of Main Computers

Assumptions for simulation model (B=Bytes b=bits)

MACHINE	I/O RATE	PACKET SIZES		
A. VAX 11/730 and VAX 11/780	.3MBps (at 1.0MBps Peak)	125B	500B	1000B
B. PE3254	1.0MBps (actually 3MBps but ETHERNET limits)		500B	1000B
C. ETHERNET	1.25MBps (includes overhead)	125B	500B	1000B

1.2.4 Data Exchanges Between Terminals and Hosts.

The data exchanges between the host VAX 11/725 or VAX 11/730 and the terminals are summarized as follows:

1. A complete data screen change will take 1920 bytes of data. (24 lines by 80 Bytes per line).

2. Screens are refreshed automatically by data transfer from the 11/780 and 3254 hosts. These automatic refreshes occur every five seconds. Each screen refresh is assumed to require 1920 Bytes of data.

3. It is possible for a terminal operator to request an update or complete screen change. These requests are initiated by a human operator who will then (usually) inspect the screen contents before making change requests. (Not all change or update requests involve a full 1920 Bytes.) It is assumed that, on the average, 3 minutes elapse between full screen change requests by a terminal operator.

4. Packet lengths are assumed to be as follows (Overhead Included):

Terminal Operator Requests: 128 Bytes/Packet (2 packets)

Screen Minor Updates or Minor Changes: 256 Bytes/packet (1 packet)

Screen Change or Refresh: 512 Bytes/packet (4 packets)
or 1024 Bytes/packet (2 packets).

5. Screen change requests by terminal operators will be assumed to be initiated in Poisson distributed time intervals with the mean time being 3 minutes.

6. The video screen requests from terminal operators are assumed to require data from either the VAX 11/780 or the PE3254, or both. As a result, 50% of the time each screen request will result in two 128 Byte packets from the host (VAX 11/730 or VAX 11/725) one packet will go to the VAX 11/780 and one to the PE3254.

Thus, screen requests will arrive from operators with Poisson arrival times with a mean value of 3 minutes. Half the requests (exact percentage will be a variable parameter) will result in data from both the VAX 11/780 and the PE3254. (Each will respond with one 1024 Byte Packet.) The other fifty percent of the requests will go to the PE3254 or the VAX 11/780 (exact split will be a variable parameter), and the response will be two 1024 Byte packets.

1.3 Ethernet Protocol.

The Ethernet original baseband version was designed, developed, and patented by Xerox and was publicly announced in 1979. Since then a cooperative effort by Digital Equipment Corporation, Intel, and Xerox has produced an updated Ethernet which is considered the standard for cable-based Local Area Networks because it is very close to the IEEE 802 CSMA/CD standard. The Carrier Sense Multiple Access with Collision Detection (CSMA/CD) control technique is the more publicized method for bus/tree topologies. The CSMA/CD broadband version was developed and patented by MITRE as part of the MITREnet Local Area Network.

1.3.1 Function and Operation.

The Ethernet is basically a multi-access, packet-switched communications channel which is managed by the control technique CSMA/CD for carrying digital data among locally distributed computing systems. A primary goal of the Ethernet specification is compatibility. In fact, Ethernet was the first to accomplish this capability.

Using the CSMA/CD control technique, each station attached to the bus must contend with the other stations to access the bus. There is no central controller which allocates access to the channel. Each station must ´listen´ (i.e. use carrier sense) to detect whether the bus is free. If another station is transmitting, a station must wait or defer its transmission until the bus is quiet. After gaining access to the bus, the transmitting station continues to monitor the medium to detect colliding transmissions on the bus. This is called ´listen while talk´ and refers to carrier detection.

1.3.2 Data Format and Structure.

Each station on the common coaxial cable must be able to transmit and receive packets with the packet format and spacing as shown in Figure 1.2 [KI86]. A packet is made up of 8 bits which equal one byte. The last bit of each byte is transmitted first, and the preamble begins a transmission. A packet may not exceed 1526 bytes or fall below 72 bytes. Included in each of these numbers is 8 bytes for the preamble, 14 bytes for the header, the data bytes, and 4 bytes for the CRC. Each field of the frame is defined as follows:

1) Preamble: 64 bits alternating 1´s and 0´s, and ending with two consecutive 1´s. Used by the receiver to establish bit synchronization and then to locate the first bit of the frame.

2) Destination Address: 48 bits specifying the station or stations which are to receive the packet. The packet may go to one station, to a group of stations, or to all stations. This is determined by the first bit: 0 - one destination, and 1 - multiple stations. If all 8 bits are set to 1, then the packet is broadcast to all.

3) Source Address: 48 bits specifying the station which is transmitting the packet.

4) Type Field: 16 bits identifying the type of higher level protocol associated with the packet. Used to interpret the following data field.

5) Data Field: 46 to 1500 bytes of data or a pad characters. A minimum combination of 46 bytes is required to ensure that the frame will be distinguishable from a collision fragment.

6) CRC - Packet Check Sequence: 32 bits containing a redundancy check. The check is defined by the generating polynomial:

$$G(x)=x^{32}+x^{26}+x^{23}+ x^{22}+ x^{16}+ x^{12}+ x^{11}+ x^{10}+ x^{8}+ x^{7}+ x^{5}+ x^{4}+ x^{2}+ x + 1.$$

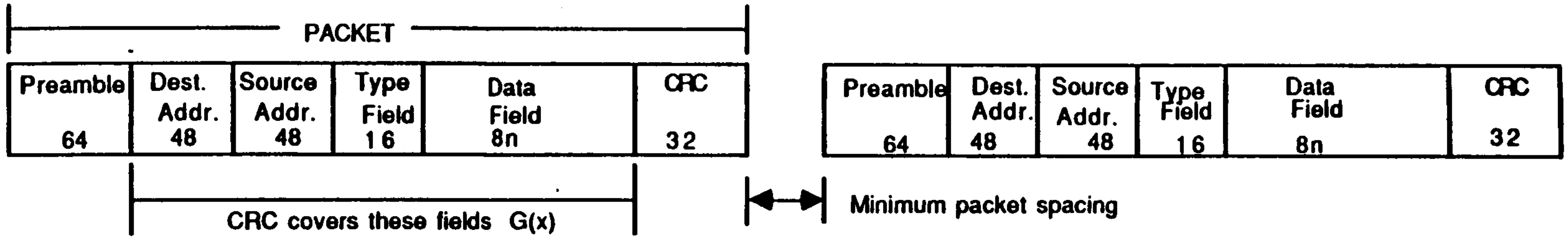

FIGURE 1.2 ETHERNET PACKET STRUCTURE

The address (destination/source), the type, and the data fields are covered by the CRC. The high-order term of the message polynomial which is divided by G(x) and produces the remainder R(x) is the first transmitted bit of the destination field. The first transmitted bit of the Packet Check Sequence field is the high-order term of R(x). A linear feedback register which is initially preset to all 1´s is used in this algorithm. After the last data bit is transmitted, the contents of this register (the remainder) are inverted and transmitted as the CRC field. After receiving a good packet, the receiver´s shift register contains 11000111000001001101110101111011($x^{31},\ldots,x^{0}$).

The Ethernet has an enforced waiting time on the bus of 9.6 micro seconds. This is the minimum amount of time which must elapse after one transmission before another may begin. It takes 51.2 micro seconds for one bit to travel from one end of the bus to the other (the round-trip propagation delay time). If any station receives a packet or bit sequence shorter than 72 bytes, the information is discarded and considered a collision fragment.

1.3.3 Hardware Characteristics.

The following three sections contain a brief overview of the hardware aspects of the Ethernet network system: channel encoding, carrier detection, and the transceivers. Additional information including detailed hardware specifications may be found in [KI86].

1.3.3.1 Channel Encoding.

The coaxial cable uses Manchester encoding which has a 50% duty cycle and insures a transition in the middle of every bit cell (´data transition´). The complement of the bit value is contained in the first half of the bit, and the second half contains the true value of the bit. (See Figure 1.3 [KI86].)

1.3.3.2 Carrier.

When data transitions are present, a carrier is present. The carrier has been lost (indicating the end of a packet) if a transition is not seen between 0.75 and 1.25 bit times since the center of the last bit cell. For purposes of deferring, the term carrier means any activity on the cable, whether properly formed or not. Any activity on either receive or collision detect signals in the last 160 nano seconds indicates carrier. (See Figure 1.3 [KI86].)

1.3.3.3 Transceiver.

At each station using the network, there are cables with taps which connect to a transceiver. The transceiver receives all signals on the cable, but only those addressed to it are received for action. The transceiver is also the device which transmits signals that are strong enough to propagate the information from one end of the cable to the other. (That is, every transmission on the cable will reach each transceiver.)

The transceiver was designed so that if it fails, the faulty device will not jam or pollute the Ethernet cable. In addition, the devices are simply built and inexpensive so that replacement of failed parts may be accomplished quickly. If a transceiver is unpowered, it disconnects itself from the cable. The transceiver also contains a watchdog timer circuit which detects incorrect behavior and shuts down the transmitter in this event. The maximum number of stations which may be attached to the cable is 1000, with the stations spaced at least 2.5 meters apart to reduce the chance that objectionable standing waves will result.

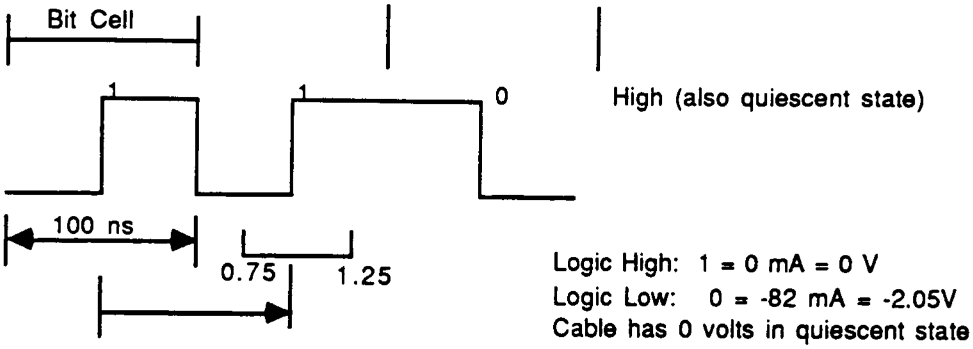
Bit Cell
1
1
0
High (also quiescent state)
100 ns
0.75
1.25
Logic High: 1 = 0 mA = 0 V
Logic Low: 0 = -82 mA = -2.05V
Cable has 0 volts in quiescent state

FIGURE 1.3 DETERMINATION OF CARRIER AT RECEIVER

1.4 Research Objective.

Within a Local Area Network environment, any of the network resources may be changed. However, the performance of the system may also change without being readily apparent. The simulation model of the Ethernet network was designed and developed in this research project to allow the user to analyze, characterize, and predict the behavior of the HOSC in a variety of scenarios. The HOSC configuration is displayed in Figure 1.1.

The first section of this report has provided an introduction to Local Area Networks, described the configuration of the HOSC, and defined the Ethernet protocol. Chapter 2 includes the definition and development of the software model and describes the user interface. A worst case analytical analysis is given in Chapter 3 and will be compared later with the simulation results. Chapter 4 discusses the outcome of several simulated scenarios. The conclusion assembles the results of this research in order to provide insight into the operations of the HOSC Video Display System using an Ethernet Local Area Network.

2.0 SIMULATION MODELING

The HOSC Ethernet Interconnection Network simulation provides the user a means of analyzing a proposed system configuration prior to hardware installation. This program provides a simulation configuration as described in section 1.2. It is assumed that there are main frames which distribute information to host computers for video display via an Ethernet bus. The activity modeled on the bus stems from the screen refresh operation (main frame to host transmission to update operator terminal), tape file dumps (main frame to host transmission to store blocks of data), and operator requests (host to main frame transmission to request a screen update).

Within the simulation, specific characteristics of the Ethernet protocol are modeled to accurately analyze the system performance. For instance, the simulation includes provisions for multiple stations trying to simultaneously access the bus by comparing the transmit times to see if they occur within a collision window. This window is actually a time period in which all stations trying to transmit within it collide. This collision happens because the signal has not had time to propagate to all parts of the network. Other stations also detect that the bus is free and begin transmission. The packets then collide. This collision window interval is calculated by using the propagation delay value between the two devices. If the transmit times occur within this window then a collision will occur. In the same fashion, stations may also have to defer a transmission if another station has gained access to the bus. A packet is deferred if the station wishes to transmit before the end of the slot time plus the propagation delay between the two devices plus the minimum delay time on the bus (9.6 microseconds). When a collision occurs, the stations involved must wait a random period of time before trying to transmit again. The Ethernet protocol specifies an exponential backoff algorithm, which is required to help minimize repeated collisions, to generate the next transmit time. The backoff number is a random number between 0 and 2^n times 51.2 microseconds,where n is the number of the current retransmit attempt and n is less than 10. (The maximum end-to-end, round-trip propagation delay for a bit is 51.2 microseconds.) In addition, a packet transmission is aborted and a jam pattern of four bytes is transmitted on the bus when a collision is detected. This jamming sequence lasts long enough so that other stations involved in the collision notice the jamming pattern. The Ethernet protocol also specifies that there must be a minimum wait time of 9.6 microseconds between any two transmissions on the Ethernet cable.

The program presented here incorporates the Ethernet protocol characteristics as stated above. The following sections describe the simulation performance parameters, how to use the program, and the software design and construct.

2.1 Performance Parameters.

There are three primary performance parameters which are of interest when analyzing a Local Area Network.

1) Throughput - The total amount of data which was actually transmitted successfully on the cable. Also defined by [ST84] as the total rate of data being transmitted between nodes (carried load).

2) Delay - The amount of time that a packet must wait between the time when the packet is ready to be transmitted at a node and the time when transmission has been completed successfully.

3) Utilization - The total amount of data (or offered load) offered to the bus presented as a percentage of bus capacity. Also defined by [ST84] as the fraction of total capacity being used.

The throughput simulation results are given as simulated throughput and theoretical throughput. Both values are presented for comparison purposes. The simulated throughput is calculated in the following manner:

$$S = \frac{U}{B + I}$$

where B = average duration of the channel busy period
I = average duration of the channel idle time
U = average time during a cycle time that the channel functions without collisions
B + I = average cycle time.

Thus, the simulated throughput is a measurement of the channel activity. The above values are tallied as the program runs and keeps accurate records of exactly what is occurring on the bus. The theoretical throughput, however, is a calculated measurement. The following formula was derived [KI86] for the CSMA protocol:

$$T = \frac{Ge^{-AG}}{G(1 + 2A) + e^{-AG}}$$

where G = number of new packets per unit of time + number of retransmitted packets per unit of time, and
A = unit of propagation time.

Thus, G is the offered load to the system - the total amount of information new and repeat which was transmitted on the channel. This value is recorded as the program runs and is used in the above equation to produce the theoretical throughput. Notice, however, that the theoretical results will be much lower than the simulated results because the CSMA protocol does not have the collision detect capability or the exponential backoff calculations (to minimize repeated collisions) as does the CSMA/CD protocol which is modeled here. The theoretical value is included as a fundamental parameter used for comparison and verification of the simulation results.

To produce the efficiency measurement, $E = \frac{S}{G}$, where S is the simulated throughput and G is the offered load as defined above. The efficiency performance figure describes the percentage of time a transmission will occur with no collision.

In addition to the performance figures discussed, each device attached to the bus has statistics which are of interest when examining the overall performance. These performance figures include 1) the total waiting time of a device due to packets being deferred from transmission, 2) the total waiting time of a device due to packet collisions, 3) the minimum amount of time a device has ever had to wait to access the bus due to a packet being deferred or being caught in a collision, and 4) the maximum amount of time the device has had to wait for a packet to be transmitted (including a collision, random backoff, and retry). A break down of the number of defers and number of collisions that a device has experienced is also included as well as a count of the number of packets it has received and transmitted. The ´max wait receiver´ data is included to indicate the device which was to receive the packet that had the maximum packet wait time associated with it. Table 2.0 (page 29) depicts a typical printout of a simulation run.

2.2 User Interface.

The user input parameters include:

1) Number of main frames and hosts attached to the bus
2) Number of terminals attached to each host
3) Data I/O rates for each device
4) Packet sizes to be transmitted on the bus
5) Number of packets to be transmitted for a screen refresh or tape file dump
6) Distance from reference point of each device attached to bus.
7) Ethernet bus I/O rate
8) Simulation run time
9) Frequency each host will be updated by main frames (screen refresh or tape file dump)
10) Frequency of operator requests (randomly occur within this time frame).

The above parameters are entered by the operator before execution of the simulation to specify the desired configuration. These are grouped into three categories: general configuration information, main frame data, and host parameters.

Upon selection of the general category, the operator enters a descriptive heading to identify the information contained in the data file selected. A random number must be entered to seed the internal random number generator. The simulation run time implies how long the operator wishes to record the activity simulated on the bus. A discussion of how long the simulation run should be is included in section 2.5. The upper bound on the number of main frames allowed by the operator is 10, on the number of hosts is 50, and the maximum number of terminals per host is 20. However, these numbers are contained in the heading of the program source code and may easily be modified by an experienced software engineer. The only limitations should be the memory on the computer system and the increased simulation run-times. Another general parameter is the frequency of the screen refresh operation and the tape file dumps. The number entered for the screen refresh indicates that the main frame will update each host within this time frame. For example, if there are two main frames and ten hosts to be sent a screen refresh every 5 seconds, then the first host will receive an update at time 0.5 seconds, the second at time 1.0 seconds, the third at time 1.5 seconds, and so forth. The same holds true for the tape file dump operation. The operator request frequency indicates that this is the average time at which an update will be requested. The request is based on a Poisson interval so that the requests occur randomly. When an operator request is performed, it is assumed within this model that each main frame will be sent one packet and in return each main frame will send the appropriate number of packets to update the screen.

For each main frame attached to the bus, the operator must enter the I/O rate of the device. The question is, ′How fast on the average is the information processed on the device and placed in buffers to be transmitted on the Ethernet bus?′ In this case, the first packet sent by a device is assumed to be buffered and ready to transmit on the bus, but the device I/O rate will determine how fast the buffer will be filled so that the next packet may be sent. The first packet will be transmitted on the bus when the bus becomes free, but the next packet may not be transmitted immediately since another device may have gained access to the bus as the second packet was being formatted for transmission. For a numerical example, let the VAX 11/780 I/O rate equal .3Mbps while the Ethernet bus is 10 Mbps, then if packet sizes are 7.68 kb, the first packet transmits in 770 microseconds, but the second packet for the VAX takes 25.6 milliseconds to format. Therefore, the dead time on the bus is 24.8 milliseconds unless accessed by another device. This discussion should help show how device I/O rates affect system performance. In addition to device I/O information for each main frame, the operator chooses some point at the end of the cable and gives the distance of each device from this point. For main frame activity, the operator enters the number of bits per packet and the number of packets to be sent for a screen refresh operation and for a tape file dump.

For each host attached to the bus, the operator specifies the number of terminals attached, the I/O rate, the number of bits per packet, and the distance from the chosen reference point. The host I/O rate is important when there are multiple main frames attached since each main frame would receive one packet from the host performing the operator request. Thus, the I/O rate of the host would determine how fast the main frame received the request and in turn would affect the rate at which the terminal received the requested update.

2.3 Simulation Software Design and Construct.

The Ethernet simulation source code is divided so that there is a short main routine which calls many subroutines to do the actual work. The primary logic is displayed in the flow chart of Figure 2.0. The program variables are set to their appropriate initial states in the subroutine Initialize. Configure is the subroutine which allows the operator to change the system configuration in order to fill his specific requirements. (See Section 2.2 for more details on the operator input parameters.) PrintConfig writes the configuration set up by the operator to a file called Auxout so that this information may be printed when the simulation has finished.

The actual simulation begins when the routine FindNext is invoked to determine the next station to transmit a packet depending on the next smallest transmit time. A call to Acollision determines if there is a collision on the bus, performs the exponential backoff algorithm, insures a jam pattern is sent, and updates the affected system parameters (wait time, system clock, etc). If there is no collision, then the Checkdefer routine determines if there are any stations which must defer its transmission until the current transmission is complete. The most complicated subroutine may be the Update routine which handles updating the station which just transmitted. This includes handling cases such as a device which has additional packets to transmit and a host which has sent an operator request forcing the main frame to send the data requested. These routines are repeated over and over until the program has performed the bus simulation as long as specified by the operator (sim time).

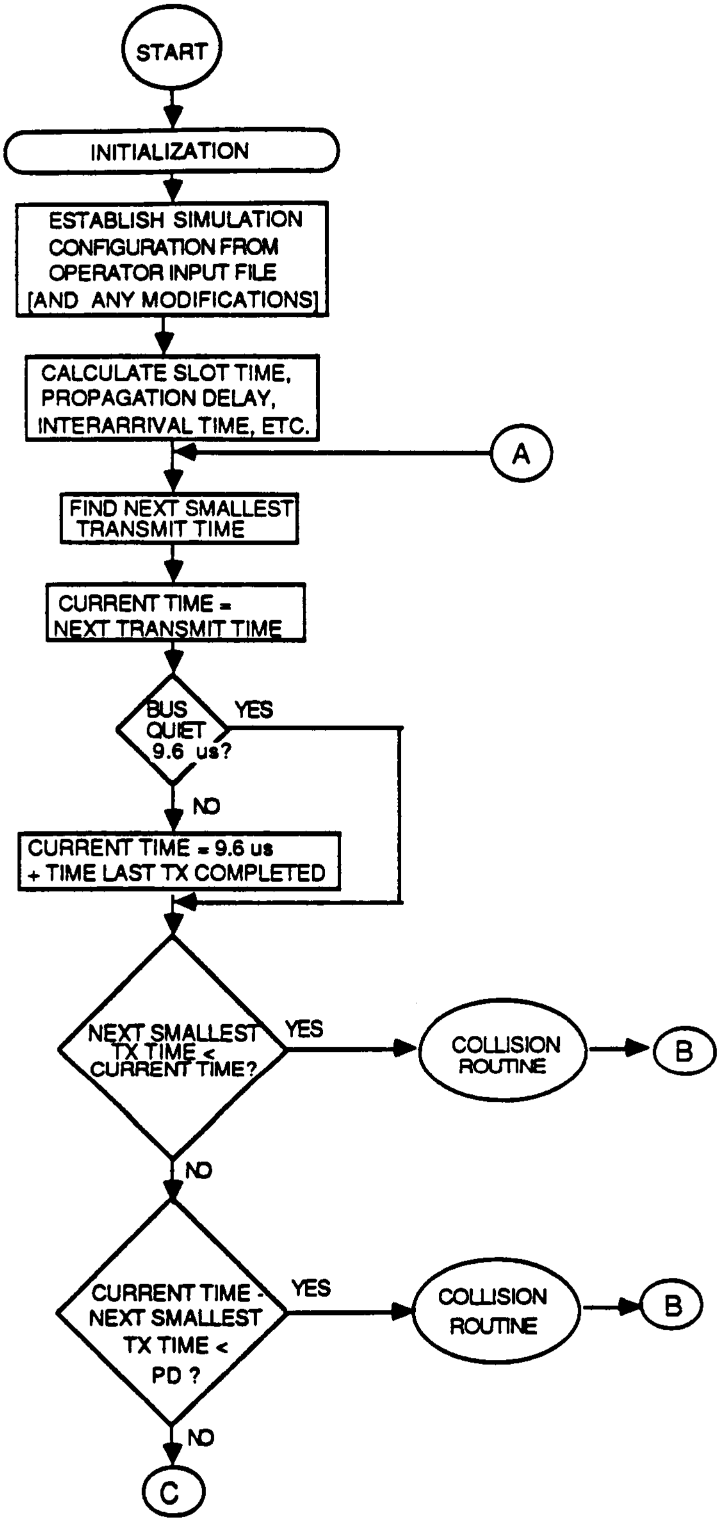
START
INITIALIZATION
ESTABLISH SIMULATION CONFIGURATION FROM OPERATOR INPUT FILE [AND ANY MODIFICATIONS]
CALCULATE SLOT TIME, PROPAGATION DELAY, INTERARRIVAL TIME, ETC.
A
FIND NEXT SMALLEST TRANSMIT TIME
CURRENT TIME = NEXT TRANSMIT TIME
BUS QUIET 9.6 us?
YES
NO
CURRENT TIME = 9.6 us + TIME LAST TX COMPLETED
NEXT SMALLEST TX TIME < CURRENT TIME?
YES
COLLISION ROUTINE
B
NO
CURRENT TIME - NEXT SMALLEST TX TIME < PD ?
YES
COLLISION ROUTINE
B
NO
C

FIGURE 2.0 FLOW CHART OF ETHERNET SIMULATION CODE

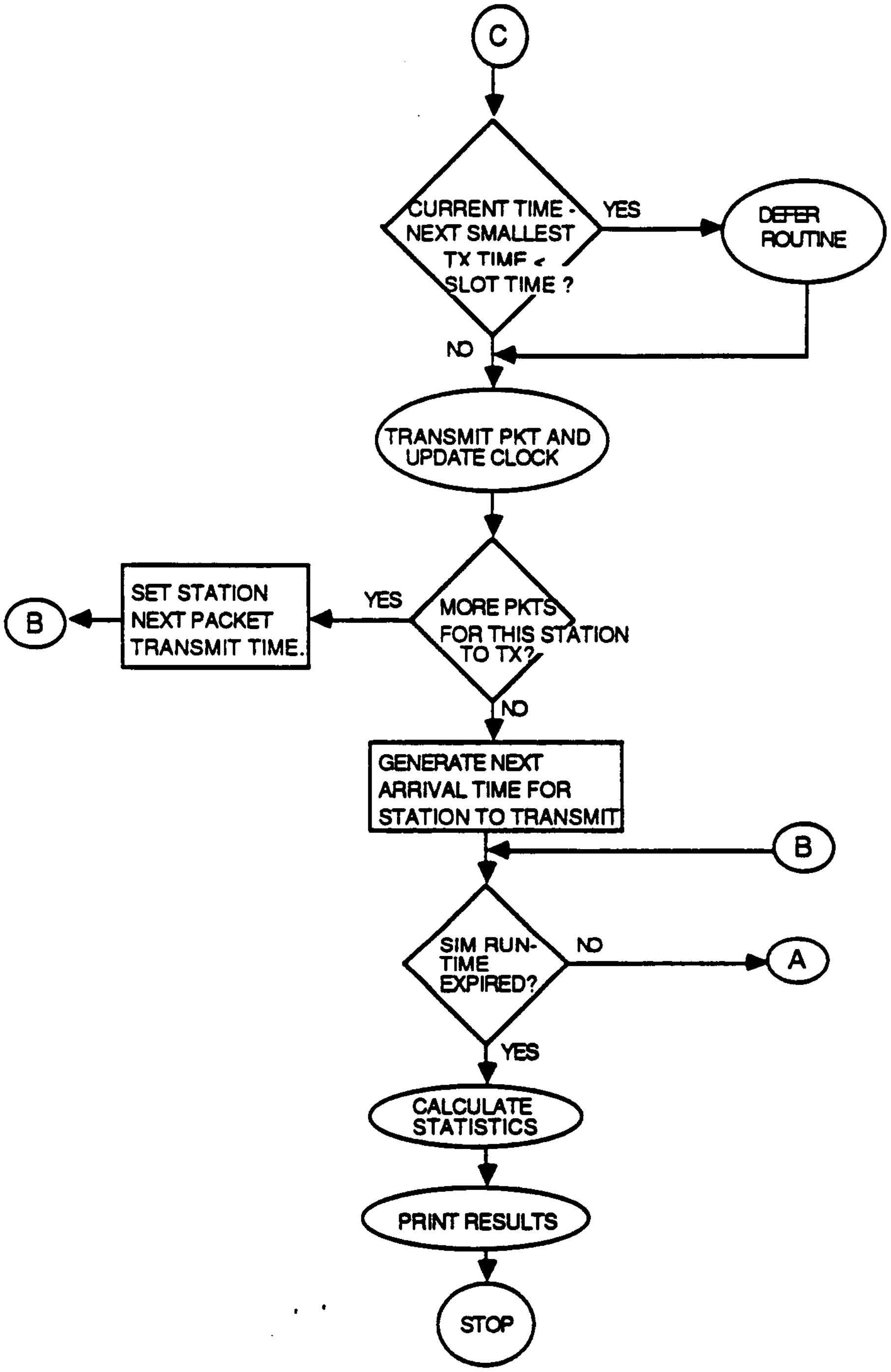
C
CURRENT TIME - NEXT SMALLEST TX TIME < SLOT TIME ?
YES
DEFER ROUTINE
NO
TRANSMIT PKT AND UPDATE CLOCK
MORE PKTS FOR THIS STATION TO TX?
YES
SET STATION NEXT PACKET TRANSMIT TIME.
B
NO
GENERATE NEXT ARRIVAL TIME FOR STATION TO TRANSMIT
B
SIM RUN-TIME EXPIRED?
NO
A
YES
CALCULATE STATISTICS
PRINT RESULTS
STOP

FIGURE 2.0 (CONTINUED)

The results of the simulation are added to the file Auxout by the subroutine Printstats. The operator in addition may add a condensed copy of the simulation results to summary tables. The ChartResults routine will add the information to the summary tables found in files SF1 and SF2.

2.4 Verification of Model.

The information shown in Table 2.0 contains a summary of ten simulations with the system configuration modified each time. The graph of Figure 2.1 plots the offered load (G) to the system versus the simulated (S) and theoretical (T) throughputs for the ten examples. The theoretical curve is identical to the plot produced in [KI86]. The equation for the theoretical throughput is given in section 2.1. However, the theoretical information only represents the CSMA protocol and not CSMA/CD. Comparing the two curves shows that the additional features - carrier detect and the exponential backoff algorithm - greatly increases the performance of the CSMA/CD protocol.

Two references [SH80] and [ABA77] also produce similar plots to show the Ethernet bus utilization. According to [ABA77], the maximum throughput rate is 80% when the bus is fully loaded. And, [SH80] specifies that the throughput rate when the offered load is 100% increases as packet sizes become larger - 512 bytes/packet = 96% throughput, 128 bytes/packet = 88% throughput, and 64 bytes/packet = 83% throughput due to the increased possibility of packet collisions. The results produced by this Ethernet simulation agree with these

TABLE 2.0
SIMULATION RESULTS OF TEN RUN SCENARIOS

SIMULATION RUN PARAMETERS

S.T. - SIMULATION RUN TIME
E.B. - ETHERNET BUS
M.F. - MAIN FRAME
T.F.D. - TAPE FILE DUMP
H. - HOSTS
T/H - TERMINALS PER HOST
S.R. - SCREEN REFRESH
O.R. - OPERATOR REQUEST

		I/O RATES			NUMBER OF			NUMBER PKTS FOR			BITS/PKT FOR			AVG TIMES (SEC)		
RUN	S.T.	E.B.	M.F.	H.	M.F.	H.	T/H	S.R.	T.F.D.	O.R.	S.R.	T.F.D.	O.R.	S.R.	T.F.D.	O.R.
1	5.0E+00	1.0E+07	1.0E+07	2.4E+06	2	20	10	2	2	1	7.7E+03	7.7E+03	1.0E+03	2.0E+00	1.5E+01	1.8E+02
2	5.0E+00	1.0E+07	1.0E+07	2.4E+06	2	20	10	2	2	1	7.7E+03	7.7E+03	1.0E+03	2.0E+00	1.0E+01	1.0E+01
3	5.0E+00	1.0E+07	1.0E+07	2.4E+06	2	20	10	2	2	1	7.7E+03	7.7E+03	1.0E+03	2.0E+00	4.0E+00	1.0E+01
4	5.0E+00	1.0E+07	1.0E+07	2.4E+06	2	20	10	4	4	1	7.7E+03	7.7E+03	1.0E+03	2.0E+00	1.5E+01	1.8E+01
5	5.0E+00	1.0E+07	1.0E+07	2.4E+06	2	20	10	4	6	1	7.7E+03	7.7E+03	1.0E+03	2.0E+00	1.0E+01	1.0E+01
6	5.0E+00	1.0E+07	1.0E+07	2.4E+06	2	20	10	6	6	1	7.7E+03	7.7E+03	1.0E+03	2.0E+00	1.5E+01	1.8E+01
7	5.0E+00	1.0E+07	1.0E+07	2.4E+06	2	20	10	6	6	1	7.7E+03	7.7E+03	1.0E+03	2.0E+00	1.0E+01	1.0E+01
8	5.0E+00	1.0E+07	1.0E+07	2.4E+06	2	20	10	8	6	1	7.7E+03	7.7E+03	1.0E+03	2.0E+00	1.5E+01	1.8E+01
9	5.0E+00	1.0E+07	1.0E+07	2.4E+06	2	20	10	8	10	1	7.7E+03	7.7E+03	1.0E+03	2.0E+00	1.5E+01	1.8E+01
10	5.0E+00	1.0E+07	1.0E+07	2.4E+06	2	20	10	10	10	1	7.7E+03	7.7E+03	1.0E+03	2.0E+00	1.5E+01	1.8E+01

SIMULATION RUN RESULTS

G - AGGREGRATE OFFERED LOAD AS A PERCENT OF BUS CAPACITY
S - SIMULATED THROUGHPUT
T - THEORETICAL THROUGHPUT
E - EFFICIENCY
TOT WAIT DEFER TIME - MAXIMUM TOTAL WAITING TIME OF A DEVICE DUE TO PACKETS BEING DEFFERED
TOT WAIT COLL TIME - MAXIMUM TOTAL WAITING TIME OF A DEVICE DUE TO PACKET COLLISIONS
MIN WAIT PACKET - MINIMUM PACKET WAIT TIME TO ACCESS ETHERNET BUS
MAX WAIT PACKET - MAXIMUM PACKET WAIT TIME TO ACCESS ETHERNET BUS

RUN	G	S	T	E	NUM COLS	PKTS TMIT	TOT WAIT DEFER TIME	TOT WAIT COLL TIME	MIN WAIT PACKET	MAX WAIT PACKET
1	2.1E-01	1.8E-01	1.7E-01	8.7E-01	182	1217	5.1E-02	5.8E-02	9.9E-06	5.9E-03
2	2.9E-01	2.3E-01	2.3E-01	8.0E-01	431	1797	8.3E-02	7.1E-02	1.7E-06	2.6E-02
3	3.2E-01	2.8E-01	2.4E-01	8.6E-01	331	2077	7.7E-02	5.2E-02	1.1E-06	2.5E-02
4	5.2E-01	4.2E-01	3.4E-01	8.1E-01	715	2933	1.8E-01	2.4E-01	1.7E-06	4.4E-02
5	6.0E-01	4.8E-01	3.7E-01	8.0E-01	903	3365	1.8E-01	1.7E-01	7.3E-07	4.7E-02
6	7.4E-01	6.0E-01	4.3E-01	8.2E-01	1022	4113	2.4E-01	5.1E-01	1.4E-06	4.7E-02
7	8.6E-01	6.7E-01	4.6E-01	7.8E-01	1372	4631	3.0E-01	6.1E-01	3.5E-07	5.0E-02
8	9.2E-01	7.4E-01	4.8E-01	8.1E-01	1301	5011	2.7E-01	1.1E+00	3.5E-07	5.2E-02
9	1.0E+00	7.9E-01	5.0E-01	7.9E-01	1568	5368	3.0E-01	1.2E+00	7.6E-08	5.1E-02
10	1.1E+00	8.5E-01	5.2E-01	7.8E-01	1742	5742	3.8E-01	1.6E+00	3.5E-07	5.2E-02

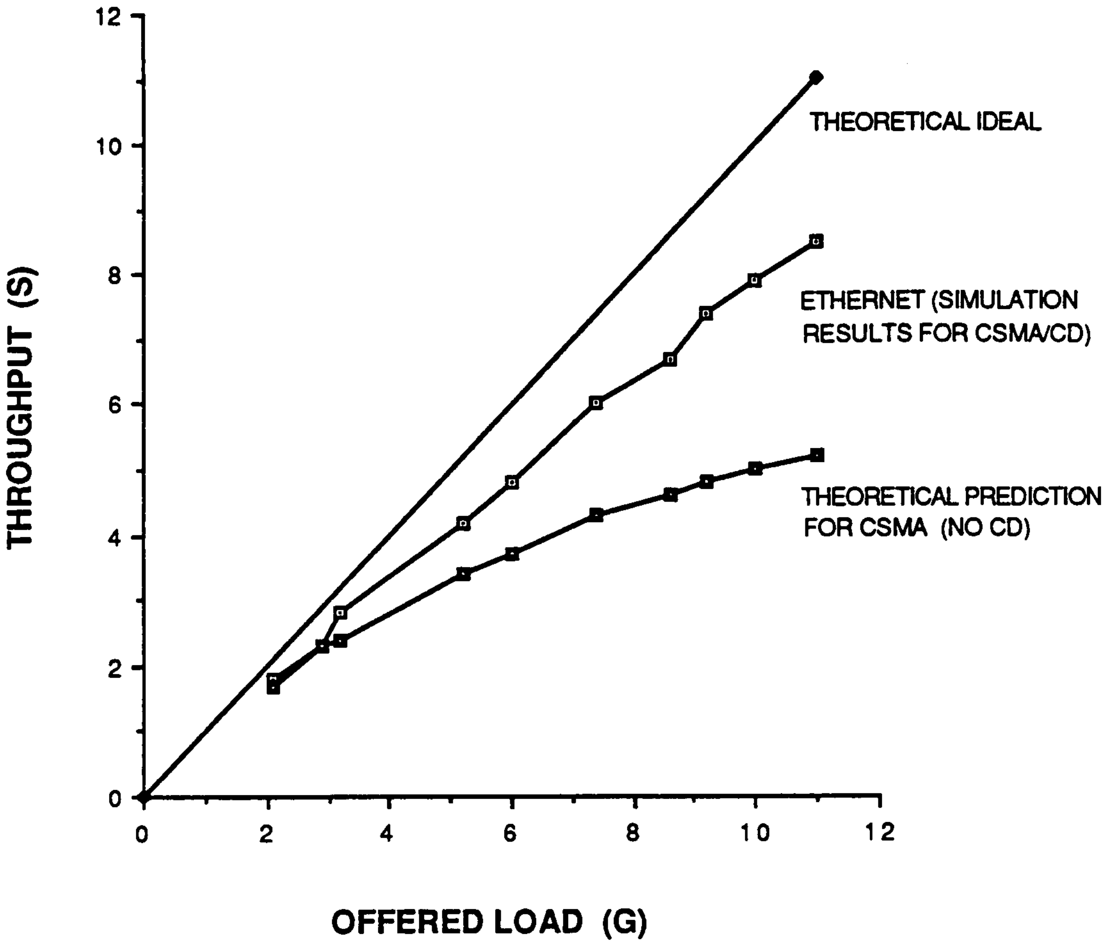

FIGURE 2.1 OFFERED LOAD VERSUS THROUGHPUT FOR TEN SIMULATION RUNS

sources since the throughput rate at 100% offered load is about 78%. This figure is almost identical to the simulation model presented by [ABA77] and is lower than the estimates of [SH80]. These sources provide a reference of comparison to verify proper operation of the Ethernet simulation.

2.5 Using the Program.

When using the simulation program, the user should have an idea of the expected results before they appear. For instance, with a lightly loaded bus the user should expect very few collisions and a very high efficiency rate. In this section, several simulation parameters will be discussed to help the user understand more fully the outcome of the simulation.

The simulation run-time is a parameter entered by the user to allow the bus activity to be observed during a certain time frame. In different situations this parameter will need to be increased or decreased depending on the activity on the bus. If, for example, the operator configures the system so that there are operator requests occurring frequently, then the amount of random information offered on the channel is high. This is a case where the run-time must be high enough to allow the bus activity to stabilize so that an accurate report is given. If the operator is unsure of this run-time figure, several runs may need to be performed and compared against each other. Notice in Table 2.1, the same configuration was run eight times with the simulation run-time increasing each time. The performance parameters: the offered load, the simulated throughput, and efficiency, remain constant for the last three runs implying that the activity on the bus has stabilized and longer simulation runs would produce the same results.

From many test simulation runs it is apparent that the Ethernet simulation requires a relatively low run-time for most configurations. This model was designed to simulate as closely as possible the identical activity of an Ethernet network. The percentage of random data greatly affects the activity on the bus since each operator request causes the main frame to stop current activity of screen refresh operations and tape file dumps and perform the requested operator request. Of course normal activity is soon resumed, but frequent random requests require the main frames to delay transmissions so that constant contention for the bus is almost guaranteed later. This in turn causes an increased collision rate and thus the efficiency of the bus is lowered due to re-try packets.

TABLE 2.1
SIMULATION RESULTS OF A CONFIGURATION MODELED OVER TIME

SIMULATION RUN PARAMETERS

S.T. - SIMULATION RUN TIME
E.B. - ETHERNET BUS
M.F. - MAIN FRAME
T.F.D. - TAPE FILE DUMP
H. - HOSTS
T/H - TERMINALS PER HOST
S.R. - SCREEN REFRESH
O.R. - OPERATOR REQUEST

		I/O RATES			NUMBER OF			NUMBER PKTS FOR			BITS/PKT FOR			AVG TIMES (SEC)		
RUN	S.T.	E.B.	M.F.	H.	M.F.	H.	T/H	S.R.	T.F.D.	O.R.	S.R.	T.F.D.	O.R.	S.R.	T.F.D.	O.R.
1	1.0E+00	1.0E+07	1.0E+07	2.4E+06	2	20	10	20	20	1	7.7E+03	7.7E+03	1.0E+03	2.0E+00	1.5E+01	1.8E+02
2	3.0E+00	1.0E+07	1.0E+07	2.4E+06	2	20	10	20	20	1	7.7E+03	7.7E+03	1.0E+03	2.0E+00	1.5E+01	1.8E+02
3	5.0E+00	1.0E+07	1.0E+07	2.4E+06	2	20	10	20	20	1	7.7E+03	7.7E+03	1.0E+03	2.0E+00	1.5E+01	1.8E+02
4	1.0E+01	1.0E+07	1.0E+07	2.4E+06	2	20	10	20	20	1	7.7E+03	7.7E+03	1.0E+03	2.0E+00	1.5E+01	1.8E+02
5	1.5E+01	1.0E+07	1.0E+07	2.4E+06	2	20	10	20	20	1	7.7E+03	7.7E+03	1.0E+03	2.0E+00	1.5E+01	1.8E+02
6	2.0E+01	1.0E+07	1.0E+07	2.4E+06	2	20	10	20	20	1	7.7E+03	7.7E+03	1.0E+03	2.0E+00	1.5E+01	1.8E+02
7	3.0E+01	1.0E+07	1.0E+07	2.4E+06	2	20	10	20	20	1	7.7E+03	7.7E+03	1.0E+03	2.0E+00	1.5E+01	1.8E+02
8	5.0E+01	1.0E+07	1.0E+07	2.4E+06	2	20	10	20	20	1	7.7E+03	7.7E+03	1.0E+03	2.0E+00	1.5E+01	1.8E+02

SIMULATION RUN RESULTS

G - AGGREGRATE OFFERED LOAD AS A PERCENT OF BUS CAPACITY
S - SIMULATED THROUGHPUT
T - THEORETICAL THROUGHPUT
E - EFFICIENCY
TOT WAIT DEFER TIME - MAXIMUM TOTAL WAITING TIME OF A DEVICE DUE TO PACKETS BEING DEFFERED
TOT WAIT COLL TIME - MAXIMUM TOTAL WAITING TIME OF A DEVICE DUE TO PACKET COLLISIONS
MIN WAIT PACKET - MINIMUM PACKET WAIT TIME TO ACCESS ETHERNET BUS
MAX WAIT PACKET - MAXIMUM PACKET WAIT TIME TO ACCESS ETHERNET BUS

					NUM	PKTS	TOT WAIT	TOT WAIT	MIN WAIT	MAX WAIT
RUN	G	S	T	E	COLS	TMIT	DEFER TIME	COLL TIME	PACKET	PACKET
1	9.8E-01	8.4E-01	5.0E-01	8.5E-01	220	1103	6.5E-02	8.6E-01	6.7E-06	5.2E-02
2	1.0E+00	9.1E-01	5.1E-01	8.8E-01	521	3593	1.2E-01	1.8E+00	2.3E-06	5.2E-02
3	1.1E+00	9.3E-01	5.2E-01	9.8E-01	937	6117	2.0E-01	2.8E+00	4.3E-07	5.2E-02
4	1.1E+00	9.3E-01	5.1E-01	9.8E-01	1810	12152	3.5E-01	4.6E+00	4.3E-07	5.2E-02
5	1.0E+00	9.3E-01	5.1E-01	8.8E-01	2536	18180	5.1E-01	7.1E+00	4.3E-07	5.2E-02
6	1.0E+00	9.3E-01	5.1E-01	8.9E-01	3254	24309	6.5E-01	9.0E+00	4.3E-07	5.2E-02
7	1.0E+00	9.3E-01	5.1E-01	8.9E-01	4712	36384	9.6E-01	1.3E+01	4.3E-07	5.2E-02
8	1.0E+00	9.3E-01	5.1E-01	8.9E-01	7527	60630	1.6E+00	2.2E+01	4.3E-07	5.2E-02

3.0 WORST CASE ANALYTICAL ANALYSIS

A worst case analytical analysis has been developed for the HOSC peripheral processing system. This analysis calculates the maximum expected waiting time for the response to a screen change request by a specific video terminal operator. This waiting time is the time period between the operators screen change request and the receipt of the new screen data (1920 bytes assumed data requirement) at the operators terminal.

A list of parameters and symbols (Table 3.0) is provided to facilitate the reading. Table 3.3 provides a summary of the analysis which was conducted for two protocols and several parameter variations.

The two protocols concern the method for transmitting the data required for performing screen refresh update (approximately 1920 bytes per terminal). The protocols are stated as follows:

Protocol 1: All screen refresh data updates or tape file data transfers are transmitted to <u>all</u> hosts in large packets and as quickly as possible with no time breaks in the data transmissions from host to host.

Protocol 2: All screen change data updates (refresh) or tape file data transfers are transmitted to a single host. An enforced waiting period is then induced on the main frame computer before the screen change data updates or tape file data transfers are for the next host are transmitted. This waiting period allows transmission of any screen change requests from the host computers to the main frame computers in a timely fashion.

As may be observed in the results in Table 3.3, protocol 2 provides a shorter maximum waiting time for response to a screen change by a terminal operator.

TABLE 3.0
PARAMETERS AND ASSOCIATED SYMBOLS

N = The number of host VAX computers.
W = The number of bytes per second for the host to video terminal I/O.
U = The number of bytes required for a complete screen change.
V = The number of video terminals attached to a host VAX.
X = The number of packets sent for a screen refresh or screen update.
I = Screen update refresh interval.
ET = Main frame enforced waiting period during screen refresh update data transfers host-to-host.
Y = The number of bytes required to request a screen change.
B = Bandwidth of ETHERNET in bits per second.
C = Collision percentage of packet transmission on ETHERNET.
MXWT(SCR) = Maximum waiting time for response to a terminal operator screen change request.
MXWT(T-H) = The maximum waiting time for a screen change request message from a terminal to be received by a host VAX.
MXWT(H-T) = The maximum waiting time for a screen change request data packet to be received by the terminal from its host.
MXWT(H-MF) = The maximum waiting time for a screen change request message from a host to be received by a main frame.
MXWT(MF-H) = The maximum waiting time for a screen change request data transmissions from a main frame to be received by the host.

3.1 Analysis

Each host is assumed to have V video terminals per host VAX 11/730 or VAX 11/725 units. Each video terminal operator is assumed to request a screen change with Poisson distribution time intervals between the screen change requests. The average screen change request is assumed to be a mean of M minutes per request.

From the material in Stuck and Arthurs text,´´A Computer and Communications Network Performance Analysis Primer´´, Page 428 [SA85], the addition of independent mixed Poisson servers with different rates (or means) results in a Poisson server with a mean equal to the sum of the means. Since each video terminal operator is assumed to operate his terminal independently of the other operators it then is feasible to estimate the overall screen change requests from a single host with video terminals as a single composite source with Poisson distributed request intervals with a mean equal to

$$M_j = \sum_{i=1}^{V} M_{ji} \qquad ,$$

where M_j is the mean screen change request time for the jth host computer (VAX 11/730 or VAX 11/725). Assuming that all operators have the same mean request time, the mean of the composite is equal to the individual means summed together.

If we have N host computers, then there are essentially N stations that could be continuously queued to transmit a screen change request packet on the ETHERNET bus at some unique time, such as after a launch anomaly occurs and all operators request some screen change simultaneously.

3.1.1 Worst Case Analysis Model

To conduct a worst case analysis, the waiting time anticipated for a video terminal screen request will be derived. This waiting time is composed of four separate entities, the waiting time operator to host, the waiting time host to main frame (PE3254 or VAX 11/780), the waiting time from main frame to host, and the waiting time host to operator. These entities will be calculated individually.

First, consider the screen change request data transfer from the terminal operator to the host (either VAX 11/725 or VAX 11/730). Assuming that the VAX host has a 1.5 M byte per second peak I/O rate on the UNIBUS and that this total bandwidth is not available for I/O to the video terminals, it will be assumed that the average I/O rate is 50 Kbytes per second both to the video terminals and to the ETHERNET data buffer. (Note the ETHERNET data buffer may be a virtual buffer, i.e.it may consist of memory in the VAX host, but it is necessary for the Ethernet controller to have a dynamic buffer which will allow data packets to be transmitted at a 10 Mbit per second (1.25 Mbyte per second) rate per packet. However, the overall multiple packet rate may be slowed to 300 Kbyte per second rate due to

UNIBUS allocation. If the UNIBUS is dedicated to I/O for the ETHERNET and has DMA, then the full bandwidth of 1.5 Mbytes per second would be available for multiple packets.

If all V terminal operators request a screen change simultaneously and a 100 byte packet is required from the screen terminal to the host to make the request, then, assuming a 50K byte I/O rate for the terminals, the longest waiting time for the screen request to be received by the host would be incurred if the other V-1 terminals received a screen refresh of approximately 2000 bytes each before the VAX host allowed the screen change request to be received. For 2000 bytes per change and V=10 this would take

$$\frac{\text{(100+2000)bytes x 9 terminals}}{\text{50 Kbytes per second}} = \text{378 milliseconds.}$$

In general, the worst case waiting time for a screen change request to be received by the host, if the host to terminal I/O rate is W bytes per second, the screen changes and refreshes require U bytes per terminals, there are Y bytes per screen change request and there are V total terminals per host, would be:

Max Waiting Time for Screen Change Request to Be Received By the Host From a Terminal Operator is

$$MXWT(T-H) = \frac{(Y+U)\ (V-1)}{W}.$$

Conversely, the fourth entity, the waiting time for the screen change request data packet to be transmitted from a host to a terminal attached to the host, would occur if the host delivered the other terminals a complete screen change first. Thus, for 10 terminals per host and 2000 bytes per screen change and a 50 Kbyte I/O rate host to terminal, the waiting time would be

$$\frac{\text{2100 bytes x 10 terminals}}{\text{50 Kbytes per second}} = \text{420 milliseconds.}$$

In general, the worst case waiting time for a screen change request to be received by the terminal from a host would be

Max Waiting Time for Screen Change Request Data Packet to be Received by the Terminal From Its Host is:

$$MXWT(H-T) = \frac{(Y+U)\ V}{W} .$$

The two entities above are independent of the ETHERNET bus since they involve only the host to terminal communications. At this point, it is necessary to use a mathematical model to estimate the worst case waiting times for host-to-main frame request and the corresponding main frame-to-host response. Several protocols can be envisioned two are discussed in the following sections.

Before continuing the analysis, it may be necessary to review. Basically, the max waiting time is composed of four entities and can be written as

$$\text{MXWT(Screen Change Request)} = \text{MXWT(T-H)} + \text{MWT(H-MF)} + \text{MXWT(MF-H)} + \text{MXWT(H-T)}$$

where the first and last terms have been developed in a worst case sense as

$$\text{MXWT(T-H)} = \frac{(Y+U)(V-1)}{W}$$

and

$$\text{MXWT(H-T)} = \frac{(Y+U)V}{W} .$$

It is now in order to develop an expression for the worst case waiting time for the screen change request to be transmitted from the terminal's host to the main frame. The worst situation arises if the host is busy receiving a screen change data set or a tape file data dump from the main frames.

Several possible protocols may be established for this main frame to host data transfer. Two are considered below.

3.1.2 Analysis Model Using Protocol 1

Protocol 1: All screen refresh data updates or tape file data dumps are transmitted to all hosts in large packets with no breaks.

Under this scenario if there are V terminals per host and N host VAX computers, then the number of bytes required per host for a screen change update would be

$$SCBPH = Vx1920,$$

and the total data to be transmitted to accomplish all screen change data transfers from main frame to host is

$$TSCB = N(SCBPH).$$

The maximum number of data bytes in an ETHERNET packet is 1500 bytes (1527 byte packet) and the minimum size packet is 46 data bytes (73 byte packet). It takes 1920 bytes to perform a single screen change hence, at least two medium length packets are required per terminal if it is not desired to construct data packet contents with data for several terminals. Two packets of length 1000 bytes each could be easily used and is assumed. Thus to service all the terminals at one host a total of 2V packets of 1000 bytes each is required. For generality, let X = the number of packets used to transmit an update or refresh to a single terminal.

With the protocol 1 as assumed it is possible that all screen change data transfers could occur before a specific terminal screen change data transfer occurs. Thus,

$$\begin{aligned} MXWT(H-MF) &= (XVN)(8x10^{-4})\text{second per packet} \\ &+ (XVN)(10;\text{sec interframe spacing}) \\ &\cong 8XVN\ 10^{-4}\text{ sec.} \end{aligned}$$

3.1.3 Analysis Model Using Protocol 2

PROTOCOL 2: All screen refresh data updates or tape file data transfers for terminals on one host are transmitted to a single host and then a waiting period is enforced on the main frame to allow for screen change requests or other messages to be transmitted. After the waiting period all terminal screen refresh data updates or tape file data transfers for a second host are transmitted etc.

With this protocol, it is obvious that the maximum waiting time would be the time required to send updates to a single host. The waiting time for transmitting screen updates to a host would be XV if X packets are used per terminal for screen refresh. A logical enforced main frame waiting time (ET) would be proportional to the screen update refresh interval (I) desired per terminal and the number of hosts (N). This would be

$$ET = I/N.$$

Thus, the maximum waiting time under protocol 2 for sending a screen change request from the host to the main frame would be

$$MAXWT(H\text{-}MF) = 8\ XV\ 10^{-4} \text{ seconds.}$$

3.2 Analysis Results

Last in the computation is the anticipated maximum waiting time for the main frame-to-host screen change request data packet transfer. Based on the protocols mentioned above this would be:

Protocol 1: $\mathrm{MAXWT(MF\text{-}H)} = 8XVN\ 10^{-4}+2\ 8\ 10^{-4} = (XVN+2)\ 8\ 10^{-4}$

Protocol 2: $\mathrm{MAXWT(MF\text{-}H)} = XV\ 8\ 10^{-4}+2\ 8\ 10^{-4} = (XV+2)8\ 10^{-4}$.

Table 3.1 summarizes the worst case waiting time for a terminal screen change request under protocol 1 or protocol 2. Table 3.2 contains the maximum waiting times for a screen change request response if all terminal operators make a request simultaneously. It is assumed that all requests are made first but that screen update refresh data transfers are not performed but rather that the screen change requests are immediately fulfilled with the requesting operator being serviced last.

To fully estimate the maximum waiting time for a screen change request to be implemented, it is in order to allow for collisions on the data transfers over the ETHERNET. Table 3.3 summarizes the maximum worst case waiting time for various collision percentages and two values for the number of video terminals, 200 and 400.

It may be observed from the data in Table 3.3 that the best performance is obtained with protocol two, that is, with a screen update refresh data transfer or tape file data dump protocol that enforces a waiting time on the main frame computers between data transfers to individual terminal host computers. Even for protocol 1 the maximum waiting time (2.373 sec.) is not a burden.

TABLE 3.1
WORST CASE MAX WAITING TIMES FOR RESPONSE TO A SINGLE TERMINAL OPERATOR
SCREEN CHANGE REQUEST
(NO COLLISIONS)

PROTOCOL 1. ALL SCREEN REFRESH UPDATES TRANSMITTED FROM MAIN FRAME TO ALL HOST TERMINALS WITH NO BREAK.

$$\mathrm{MXWT(SCR)} = \frac{(Y+U)(V-1)}{W} + \frac{(Y+U)V}{W} + (XVN+2).8.10^{-4} + 8XVN.10^{-4} = \frac{(Y+U)}{W}(2V-1) + 16(XVN+1).10^{-4}$$

IF N = 20 HOST VAX COMPUTERS, V = 10 TERMINALS PER HOST, X = 2 PACKETS PER SCREEN CHANGE.

U = 2000 BYTES PER SCREEN CHANGE, W = 50 KBYTES PER SECOND HOST TO TERMINAL I/O

Y = 100 BYTES PER SCREEN CHANGE REQUEST.

THEN MXWT(SCR) = 1.4396 SECONDS FOR W = 50 KBYTES PER SECOND. (H-T)I/O

MXWT(SCR) = 1.0406 SECONDS FOR W = 100 KBYTES PER SECOND. (H-T)I/O.

TABLE 3.1 (CONTINUED)

PROTOCOL 2. ALL SCREEN REFRESH UPDATES TRANSMITTED BY HOST WITH ENFORCED MAIN FRAME WAITING TIME BETWEEN SERVICE TO HOSTS.

$$\mathrm{MXWT(SCR)} = \frac{(Y+U)(V-1)}{W} + \frac{(Y+U)V}{W} + XV.8.10^{-4} + (XV+2)8.10^{-4} = \frac{(U+Y)}{W}(2V-1) + 16(XV+1).10^{-4}$$

FOR THE NUMBERS USED ABOVE

MXWT(SCR) + 8316 SECONDS   FOR W = 50 KBYTES PER SECOND (H-T)I/O

MXWT(SCR) + .4326 SECONDS   FOR W = 100 KBYTES PER SECOND (H-T) I/O

TABLE 3.2

WORST CASE MAX WAITING TIMES FOR RESPONSE IF ALL TERMINAL OPERATORS MAKE SCREEN CHANGE REQUEST SIMULTANEOUSLY

PROTOCOL 1. $$\text{MXWT(SCR)} = \frac{U(V-1)}{W} + \frac{UV}{W} + \frac{Y(V-1)}{W} + \frac{Y(V)}{W} + \frac{YNV8}{B} + (XVN+2)8.10^{-4} + 8XVN10^{-4}$$

$$= \frac{(U+Y)(2V-1)}{W} + 16(XNV+1).10^{-4} + \frac{8YNV}{B}$$

FOR N = 20, V = 10, X = 2, U = 2000, W = 50 KBYTES PER SECOND AND Y = 100 BYTES

MXWT(SCR) = 1.4556 SECONDS, W = 50 KBPS

MXWT(SCR) = 1.0566 SECONDS, W = 100 KBPS

PROTOCOL 2. $$\text{MXWT(SCR)} = \frac{U(V-1)}{W} + \frac{UV}{W} + \frac{Y(V-1)}{W} + \frac{Y(V)}{W} + \frac{8NYV}{B} + XV.8.10^{-4} + (XV+2)8.10^{-4}$$

$$= \frac{(U+Y)}{W}(2V-1) + 16(XV+1)\ 10^{-4} + \frac{8YNV}{B}$$

FOR THE ABOVE NUMBERS

MXWT(SCR) = .8816 SECONDS, W = 50 KBPS

MXWT(SCR) = .4822 SECONDS, W = 100 KBPS

TABLE 3.3

WORST CASE MAX WAITING TIMES FOR RESPONSE IF ALL TERMINAL OPERATORS MAKE SCREEN CHANGE REQUESTS SIMULTANEOUSLY. COLLISIONS ASSUMED (1 COLLISION RESULTS IN 2 EXTRA PACKET TRANSMISSIONS. HENCE 10% COLLISIONS RESULTS IN 20% EXTRA TRANSMISSIONS ON THE ETHERNET BUS.)
BASIC PACKET SIZE = 1000 BYTES, 2 PACKETS REQUIRED PER SCREEN CHANGE.

PROTOCOL 1:

$$MXWT(SCR) = \frac{U(V-1)}{W} + \frac{UV}{W} + \frac{Y(V-1)}{W} + \frac{Y(V)}{W} + (\frac{YNV.10^{8}}{B} + XVN+2)8.10^{-4} + 8XVN.10^{-4})(1+2C)$$

$$= \frac{(U+Y)(2V-1)}{W} + (16(XVV+1)10^{-4} + \frac{8NVY}{B})(1+2C)$$

V=10, X=2, U=2000, Y=100 BYTES, B=10MBITS PER SECOND,

BASIC TIME TO SEND 1 PACKET OF 1000 BYTES = $8X10^{-4}$ SECONDS

	N=20 HOST VAX´S (200 TERMINALS)				N=40 HOST VAX´S (400 TERMINALS)			
MXWT(SCR):	C=0%	1%	5%	10%	C=1%	1%	5%	10%
W=50KBps	1.455sec.	1.459sec.	1.522sec.	1.587sec.	2.112sec.	2.137sec.	2.242sec.	2.373sec.
W=100KBPS	1.057sec.	1.070sec.	1.122sec.	1.188sec.	1.713sec.	1.739sec.	1.844sec.	1.975sec.
W=200KBPS	.857sec.	.870sec.	.923sec.	.988sec.	1.513sec.	1.539sec.	1.644sec.	1.776sec.

TABLE 3.3 (CONTINUED)

PROTOCOL 2:

$$MXWT(SCR) = \frac{U(V-1)}{W} + \frac{UV}{W} + \frac{Y(V-1)}{W} + \frac{Y(V)}{W} + (\frac{8NYV}{B} + XV8.10^{-4} + (XV+2)8.10^{-4})(1+2C)$$

$$= \frac{(U+Y)(2V-1)}{W} + (16(XV+1)10^{-4} + \frac{8YNV}{B})(1+2C)$$

V=10, X=2, U=2000, Y=100 BYTES, B=10MBITS PER SECOND,

BASIC TIME TO SEND 1 PACKET OF 1000 BYTES = 8.10^{-4} SECONDS

	N=20 HOST VAX'S (200 TERMINALS)				N=40 HOST VAX'S (400 TERMINALS)			
MXWT(SCR):	C=0%	1%	5%	10%	C=0%	1%	5%	10%
W=50KBps	.848sec.	.848sec.	.853sec.	.878sec.	.863sec.	.865sec.	.870sec.	.877sec.
W=100KBPS	.449sec.	.450sec.	.453sec.	.459sec.	.464sec.	.466sec.	.471sec.	.477sec.
W=200KBPS	.249sec.	.250sec.	.254sec.	.259sec.	.265sec.	.266sec.	.271sec.	.278sec.

4.0 SIMULATION RUN RESULTS

In this section, the focus will be the on interpretation of the results produced by the Ethernet simulation model. This discussion will include an analysis of a specific run to explain performance parameters and the effect of varying user inputs, a configuration summary of ten runs with an intuitive preliminary analysis to predict results, a comparison of ten simulation runs to demonstrate the system response in various configurations, and a summary of the configuration limitations encountered when using an Ethernet Local Area Network.

4.1 Analysis of A Simulation Run.

A typical configuration for an Ethernet simulation run is displayed in Figure 1.1. The results of this run are shown in Appendix II, Run 9 (page 164) and will be discussed throughly. The output report of each run is divided into 5 sections:

1) Overall System Parameters
2) Specification of Main Frame Parameters
3) Specification of Host Computer Parameters
4) Summary Table of Main Frame and Host Activity on the Bus
5) Overall System Performance Results.

The first three sections display the operands entered by the user. The overall system parameters lists the Ethernet bus I/O rate, the number of main frames attached, the number of host computers attached, the number of seconds between screen refreshes for any host, the number of seconds between tape file dumps, and the average number of seconds which will pass between operator requests from a terminal.

The information as entered by the user is output for each main frame attached to the bus. The parameters include the I/O rates, the distance of the device from a reference point, the number of packets to be sent to a terminal and the number of bits per packet for a screen refresh operation and for a tape file dump. Similarly, the user inputs for each host are output. These parameters include the number of terminals per host, the I/O rate, the number of bits per packet, and the distance of the device from a reference point. In addition, the slot time is displayed for both main frames and hosts. This is the time required for one packet to be transmitted on the bus and is calculated by the bits per packet divided by the Ethernet bus data rate.

Within the report generated by the simulation program, a table (page 169) is provided which displays the activity of each device attached to the Ethernet bus. The far left column labeled SOURCE defines the device under observation. The main frames are listed first with the main frame number followed by the operation it performed. In this case, there were two main frames, each having a refresh operation (1) and a tape file dump operation (2). Thus, the 2,2 represents the second main frame entered by the operator and the tape file dump operation. Following the main frame information, the host parameters are output.

For each device attached to the bus, several important values are tallied in order to observe the Ethernet bus activity. The WAIT TIME DEFER column specifies the total amount of time the device had a packet to transmit but had to wait until the bus was free. Similarly, the WAIT TIME COLLISION column records the total amount of time the device was involved in a collision. This total includes the time during packet transmission as the collision occurred, the time spent sending the jam sequence upon detecting a collision, and the time the device was required to wait until it could retry the packet. The devices which transmit most frequently will also be the devices most likely to have a high wait time due to collisions. In this case, the two main frames transmit often, and main frame 1 experienced the greatest waiting time of 1.2 seconds.

The DEFER COUNT and COLL COUNT columns total, respectively, the number of times that a device had to wait to transmit and the number of times it was involved in a collision. For each collision, the packets of information must be retransmitted later. The total number of packets, then, is the total number of packets transmitted successfully. The number of packets that the device transmitted within the simulation time frame is in the PKTS TX column. The number of packets that a device received is also tallied in the PKTS RX section. The MINIMUM PKT WAIT TIME gives the minimum amount of time that any packet had to wait before transmission could begin, while the MAXIMUM PKT WAIT TIME displays the longest time any packet had to wait to access the bus, either because of a defer condition or because the packet collided with another transmission on the bus and had to wait before retransmitting. Finally, the MAX WAIT RECEIVER indicates the receiving device number when the maximum packet waiting time occurred. The main frames in this case are the devices greater than the number of hosts.

With 20 hosts attached, main frame number 1 is shown in this column as 21, etc. This table displays the device activity necessary for analyzing the activity on the Ethernet cable.

The overall results of the simulation are then given (bottom of page 169). The simulation run time is displayed first. Then the total number of collisions which occurred on the bus and the total number of packets which were transmitted successfully are listed. The TOT BUSBUSY figure is the total time the bus was active, including, the time used transmitting successful packets, the time used on the bus during collisions, and the minimum required dead time of 9.6 micro seconds on the bus. The TOT USAGE TIME is the total time spent transmitting packets of information on the bus without collisions. The TOT IDLE time is the amount of time the bus was unused or idle. Then the busbusy time + the idle time = the simulation time of the bus. The average values are the totals divided by the total number of packets transmitted.

The simulated throughput is calculated using values totaled within the program. This value is calculated by dividing the average usage time by the sum of the average busbusy time and the average idle time to give the successful utilization of the bus. This value may also be calculated by taking the total number of bits successfully transmitted and dividing it by the Ethernet bus rate times the simulation run time. Thus, the usage time is simply the number of successfully transmitted bits divided by the Ethernet bus rate (or the sum of the slot times for all packets transmitted successfully). For this example, the total number of packets transmitted by the main frames was 5118 at 7680 bits each, and the total number of packets transmitted by the hosts was 250 at 1024 bits each. Then, the throughput = $\frac{(5118*7680 +250*1024)}{10E6 * 5}$ = .791245. This is the same throughput calculated from values tallied by the program. The aggregrate offered load is the total amount of data offered to the bus to be transmitted. This includes jam bits transmitted and the packets involved in collisions. The offered load then may be calculated by summing the number of bits offered to the bus. In this case, there were 5118*7680 + 250*1024 + 1568*6500 (estimated since some packets were 7680 bits and some were 1024 bits) = 49,754,240 bits. The offered load is approximately 49,754,240 / (5 * 10E6) = .9951 which is very close to actual calculated value of .9964. The theoretical throughput is calculated using the formula shown in section 2.0. It displays the theoretical value for a CSMA protocol. The efficiency of the network in the configuration specified is calculated by dividing the simulated throughput by the offered load. This value specifies how efficient the network will be at transmitting the information across the bus, for example, 79% of the time the packet offered to the bus will be transmitted with no collision occurring. Another way to estimate efficiency is to calculate this percentage based on the actual number of packets transmitted and the number involved in collisions. In this case there were 5368 packets transmitted successfully and 1568 collisions. Therefore, the efficiency is about $\frac{5368}{5368 + 1568}$ = .77393, and 22.6% of the time a packet will collide with another packet.

A summary of the input and output information may be included in a special report if the user chooses this option. In this way the information from several runs may be examined from summary charts. Not all of the charts are inclusive, and the report for each run must be examined for variable data such as host I/O rates and packet sizes. However, the summary charts are very helpful when comparing the results of several simulation runs. (See Section 4.3.)

Using the results presented in the summary charts, the worst case amount of time any operator must wait for a screen update due to an operator request will be calculated in several ways. The simulation program may be used to derive the host to main frame time and the main frame to host time. The host to terminal and terminal to host times may be computed by hand. The sum of these values will give the total operator worst case waiting time to receive a screen update. Overall, the maximum packet wait time was .051 seconds. The operator sent 2 packets - one to each main frame. Each main frame then sent the terminal 4 packets. If we assume in the worst case that each packet was delayed .051 seconds, then the total time for an operator to receive the screen update is calculated as follows:
No. packets * [Max Pkt Wait + Bus Slot + (Buffer Fill Time - Bus Slot)] or (No. packets) * [Max Pkt Wait + Buffer Fill Time].

In this case, the number of packets at 1024 bits is 2 and the number of 7680 bit packets is 8. Then the total wait time is: 2*[.051+1024/2.4E6] + 8*[.051+7680/10E6] = .516997 seconds. Note that the .051 second wait time is the absolute worst case of any packet on the bus during the 5 second interval. The value is very high (in an electrical sense, not in an operator sense) since the bus is fully loaded. The same packet may collide several times, causing a very high maximum packet waiting time. Each packet will not have to wait .051 seconds, but is included here as an absolute worst case estimate. If the minimum packet wait time and the maximum packet wait time are averaged, then the average packet wait time is (7.6E-8 + 5.1E-2)/2 = .0255 seconds. Then the total refresh wait time based on averaging the maximum and minimum packet wait times is .26199 seconds. This is still a very long wait time (again, electronically but not in an operator sense) for the operator to wait for a screen refresh and actually a poor estimate of the amount of wait time.

The refresh time will now be calculated in a more accurate manner by examining the device total wait times for screen refreshes. The chart showing the device activity for this run indicates that main frame one spent 1.447 seconds waiting to access the bus in 5 seconds for 2220 packets which means that the average packet wait time is 1.447/2220 = .0006519 seconds. Main frame two spent 1.0797 seconds in 5 seconds for 2298 packets. The average packet wait time for this device was 1.0797/2298 = .0004698 seconds. Host 17 spent the most time waiting to access the bus with a time of .38552 seconds with 12 packets. The refresh required [2*(1024/2.4E6+.03274) + 4*(7680/10E6+.0006519) + 4*(7680/10E6+.0004698)] = .07696 seconds. Obviously, if a screen refresh took this long then only 65 refreshes would be accomplished in 5 seconds. But, 5368 packets were transmitted on the bus and 600 were for tape file dumps, leaving 4768 packets for operator requests and screen refreshes. If it takes 10 packets to fulfill an operator request, then about 400 requests at 63,488 bits each could be filled in 5 seconds.

The collision rate was previously calculated at 22.6%. If this value is used instead and doubled, then we may compute the total wait time as 1.452 * [2 * 1024/2.4E6 + 8 * 7680/10E6] = .01016 seconds. Thus, the new estimate is a much more realistic representation of the bus activity and the amount of time a screen refresh will actually take. (400 requests at .01016 seconds each is about 4.064 seconds while 75 tape file dumps require 1.452 * (8 * 7680/10E6) * 75 = .669 seconds.)

The terminal to host transmission required 1 transmission of approximately 64 bytes at 50 Kbytes/second, and the host to terminal update required approximately 80X24 bytes to fill the terminal. The total time for host to terminal I/O is assumed to be 39.68 milli seconds. The operator can expect to wait .01016 + .03968 = .04984 seconds for an operator requested screen refresh. If each host has 10 terminals, then 10 * .04984 = .4984 seconds are required in the worst case for each terminal to be updated. However, the amount of information sent for each terminal is very high: 8 packets at 7680 bits each (61,440 bits of data). This is an extreme exaggeration of the number of bits which would realistically be sent since only 1920 bytes or 15,360 bits are needed for a terminal update.

The objective of this simulation run was to saturate the bus causing the greatest number of collisions and thus the maximum waiting time for any packet to be transmitted on the bus.

4.2 Simulation Runs - Configuration and Expected Results.

Many simulation runs have been performed to date. The summary information of ten runs are included in Appendix II, page 176. The basic configuration consists of 2 main frames each at 10 Mbps and 20 hosts at 2.4 Mbps with 10 terminals each. For screen refreshes and tape file dumps, the packet size was 7680 bits. For operator requests, the packet size was 1024 bits. The varied parameters were the number of packets which were sent for screen refreshes and tape file dumps. Other varied parameters were the average time between tape file dumps and between operator requests. Using this configuration as the basic scenario for observations, the bus performance can be analyzed using this simulation program.

Intuitively, as the amount of information pushed onto the bus increases, there will be a greater probability of collisions between packets since many devices will begin trying to access the bus at the same time and more frequently. Indeed, as the number of packets of information sent on the bus increased, the bus became much more heavily loaded and many more collisions did occur. As the number of collisions became larger and larger, the bus became heavily saturated. More packets were offered but fewer escaped a collision. The system then remained occupied in an attempt to clear the backlog of packets. The efficiency of the system will decrease as packets are offered and the heavily loaded system will not be able to transmit the information because of the load created by many collisions that cause repeated packet transmission.

Metcalfe and Boggs [MB76] of Xerox Palo Alto Research Center used an experimental Ethernet system to analyze performance. Their system was consistently 95% plus efficient when packet sizes were above 4000 bits. They conclude, ´For packets with a size approaching that of a slot, Ethernet efficiency approaches 1/e, the asymptotic efficiency of a slotted Aloha network´. The slot is defined here as ´the maximum time between starting a transmission and detecting a collision, one end-to-end round trip delay´. Since this delay time is 51.2 micro seconds, then the smallest packet sizes (around 576 bits) will produce the least efficient network. <u>The 10 Mbps Ethernet bus would have an effective data rate of only 3.68 Mbps as packet sizes became small and the number of stations became large</u>.

4.3 Comparison of Results when Parameters Vary.

In addition to obtaining detailed results from each run, the operator may choose to have the simulation information included in summary tables containing information from previous runs. This provides a concise summary of each simulation and allows easy comparison of the results obtained from each run. In this manner, the system may be readily analyzed to determine the most acceptable configuration for an application.

The results of the ten simulation runs are included in Appendix II. The offered load was increased from 21% to 110% by varying the number of packets transmitted on the bus during refresh and tape file dump operations. The frequency of random operator requests and tape file dumps were increased to achieve the loading of the ten runs shown. For this case the throughput increased from 18% to 85%, and the bus efficiency dropped from 87% to 78%. The throughput rates were achieved in the 5 second interval as packet transmission rose from 1217 to 5368 at about 7000 bits per packet. The results of these runs are displayed in Figure 2.1 which shows the throughput rate versus the offered load. Even a heavily loaded Ethernet performs well due to exponential backoff and collision detection features.

The simulation of Appendix II, Run 11 shows the performance results when many stations attached to the bus transmit randomly. This run configuration had 20 hosts with 15 terminals each. Each main frame was set to respond to an operator request with one packet. The random operator requests occurred frequently --on the average, the requests came from each terminal within a one second time frame. This produced a 100% offered load, but the throughput was only 54%. The decrease in efficiency can be attributed to several factors: the random generation of packets offered to the bus, the increase in the number of small packets transmitted (1024 bits from a terminal to each main frame), and the number of terminals generating the random requests increased (300 terminals). The collision rate was extremely high since many packets were offered simultaneously. Therefore, during the 5 second interval, about half of the time (2.7 seconds) was spent transmitting packets with no collisions. The efficiency of the system decreased as the number of collisions increased. The effective data rate was lowered from 10 Mbps to 5.4 Mbps.

4.4 Limitations on Configuration.

Ethernet specifications limit the cable length to 2.5 kilometers. This allows a maximum of 1024 stations connected 2.5 meters apart. The simulation does not handle this number of stations. The limiting factors are the computer memory and the amount of time required to run a simulation of this magnitude. For the allowable number of stations, see section 2.2. Another specification of Ethernet is packet size. The maximum allowed size is 1526 bytes, and the minimum packet size is 72 bytes.

From the simulation runs presented, the system configuration may be varied in many ways. However, the performance of the network will vary greatly depending on how it is structured. To ensure that the system performance is high (efficiency above 75%), keep the packet size large (above 4000 bits). This will reduce the number of collisions on the bus and thus the amount of repeated data. With very small packet sizes and a high number of stations transmitting, the efficiency is reduced to 36.8% of capacity.

A high performance Ethernet is also achieved when the data is divided over time so that transmissions are structured. For example, the screen refresh and tape file dump are divided over time for each device. In 2 seconds each device received a screen update. With 20 hosts attached, a screen update was transmitted every .1 second. The data offered to the bus was not randomly generated but structured so that fewer collisions would occur.

Another limiting factor is the data rate of the devices attached to the bus. The bandwidth of the Ethernet cable is 10 Mbps. However, there is no way to utilize the maximum bandwidth unless there are devices attached which can supply information to be transmitted at that rate. In the example runs of Appendix II, the two main frames were also modeled with a bandwidth of 10 Mbps. If the data rate of one of the main frames had been much less than this, then the total configuration of the system would have changed. The offered load of the system should be much less since the main frame would not be able to offer the same amount of data to the bus at the same rate. Using the example from section 4.1 and changing only the data rate of main frame 1 to 2.4 Mbps, the offered load dropped to .775 with a throughput rate of .594 (compared with the offered load of 1.0 and throughput rate of .79 for the same configuration). Thus, the VAX 11/780 with a data rate of 2.4 Mbps will be a limiting factor in the utilization of the bus since less information will be offered in the same time frame.

5.0 SUGGESTIONS FOR HOSC SYSTEM RESPONSIVENESS

Several considerations for improving system responsiveness are presented. These may be separated into two basic categories: system protocol for data transmissions and packet sizes for control and data transmissions.

5.1 Protocol

Two protocols are mentioned in sections 3.1.2 and 3.1.3. They are restated below for convenience:

Protocol 1: All screen refresh data updates or tape file data transfers are transmitted to all hosts in large packets with no breaks. Thus, all hosts are updated at essentially the same time.

Protocol 2: All screen change data updates or tape file data transfers for terminals on one host are transmitted to a single host, and then a waiting period (or break) is enforced on the main frames to allow for screen change requests or other messages to be transmitted. After the waiting period, all terminal screen refresh data updates or tape file data transfers for the second host are transmitted, etc.

The second protocol is recommended for optimum system performance. It may be noted from the data of Table 3.3 that either protocol will function satisfactorily but the second protocol will ensure that timely action will be performed on a screen request from a terminal operator by ´allowing the request time slots´.

5.2 System Considerations

Other system considerations revolve around timing and packet structure. Analysis and Simulation have shown that screen refresh update transmissions (about 2000 bytes each screen) and tape file data dumps (about 2000 bytes per screen) dominate the ETHERNET data bus activity. It is desirable to eliminate as much of this activity as is feasible. One way to reduce data flow would be to send only the new data required per screen (or changed parameters) from the PE3254 and VAX 11/780 to the various terminal host computers (the VAX 11/730 and/or VAX 11/725). By creating a fixed format for each screen display and identifying those parts of the fixed format that are not variables, only the dynamic data need be transmitted from the PE33254 and VAX 11/780 to the host computers. The host computers would have permanent prestored formats for each screen display. This would allow shorter packets to be transmitted from the PE3254 and VAX 11/780 thus reducing bus busy time.

To reduce the communication required for a screen change request it is desirable to use an address code and data field for the screen change request that is used by both the VAX 11/780 and the PE3254. That is one packet transmitted but received by both units with appropriate data field instructions for each unit. Thus, a screen change request from one video terminal operator results in one packet transmitted from its host. This one packet carries a unique address which is recognized by both the VAX 11/780 and PE3254 and the data field has the request data which allows both the VAX 11/780 and the PE3254 to respond.

An effective speed up in screen change request action may be achieved by forcing the VAX host to allow screen change request messages from the video terminals to be received in between screen change or refresh data packets being transmitted from the host to the other terminals attached to that host.

Data transmissions from the main frames to the various hosts must be structured so as to be packetized. If these packets are very long, the main frame tends to tie up the ETHERNET data bus for long periods of time. Thus, an upper bound on the packet size is desirable. On the other hand, many short packets tend to raise the packet collision frequency therefore, it may be desirable to lower bound the packet size. A good packet size is probably lower bounded by 2000 bytes and upper bounded by the total data per host about 20,000 bytes. Since ETHERNET packets are limited to an upper bound of approximately 1500 data bytes per packet it is desirable to allocate two 1000 byte packets per screen change or screen refresh or tape file data transfer per screen.

6.0 CONCLUSION

The purpose of this research has been to provide a software tool that will aid in the performance analysis of an ETHERNET system in varying configurations. This program was specifically designed to meet the needs of the engineers and scientists at NASA's Huntsville Operation Support Center. The model provides for main frame computers which primarily distribute satellite data to host computers as well as the random operator requests for terminal updates.

The study began with a description of the system configuration and operation to be modeled. Figure 1.1 gives an overview of the system configuration. This establishes a basic configuration for test cases in Section 4.0. The device characteristics are provided in Section 1.2, which serve as the components of the network modeled. A complete description of the Ethernet protocol is then presented in Section 1.3 to provide insight into Ethernet characteristics and operations.

Section 2.0 was provided to explain the performance parameters used to analyze the network and to describe how to use the simulation model. This section also provides the validation criteria used for this program. Using the results of many simulation runs, the offered load versus throughput rates were plotted in Figure 2.1. The graph produced results very close to the research efforts of [ABA77] and [SH80]. For a heavily loaded bus with 22 devices attached and packet sizes approximately 7000 bits, the bus efficiency was near 80%. The bus utilization then was about 8 Mbps although under certain conditions the utilization may reach 98% [MB76].

From Section 3.0 a worst case analytical model was presented to determine the maximum waiting time of an operator requested screen update. Protocol 2 corresponds more closely with the simulation model since main frames have structured times for screen refreshes (see Section 4.4). These results show that for a host to terminal data I/O rate of 50 Kbps and a system having 20 hosts with 10 terminals per host, the operator would wait .878 seconds if we assume a 10% collision rate. In comparison, the simulation model produced the results of Section 4.1. From these four calculations of round trip host to main frame calculation, the results were .516997, .26199, .07696, and .01016 seconds. The largest was calculated using the longest packet waiting time during the entire simulation run. The shortest time was computed based on the collision rate of 22.6% doubled. Adding the host to terminal I/O to the above values produced .556677, .32494, .11664, and .04984 seconds. The wait time of .04984 seconds was shown to be the most realistic wait time that an operator would experience. As expected, the simulation model demonstrates a much shorter operator wait time than the absolute worst case analytical model even on a heavily loaded bus.

In Section 4.3 the results of several simulation runs were presented. In summary, the throughput of the network will approach 98% under favorable conditions. However, adverse conditions such as small packet sizes combined with a high number of stations produce a lower bound of

36.8% channel throughput. The examples within this report average a throughput rate of 80% demonstrating the robust features of Ethernet. Causing much of the data to be randomly generated (not time structured transmissions) will also reduce the Ethernet throughput rate since this will produce a higher collision count.

Four suggestions in Section 5.0 were proposed to reduce inefficiency within the network: eliminate unnecessary data transmissions by restructuring system operations, broadcast messages from host to main frames for operator requested screen updates, hosts must be able to handle more than one terminals activity at a time, and maintain packet sizes of about 1000 bytes each.

The accuracy of this software model is limited by the understanding gained during research. But, it has been validated by comparing theoretical expected results to those achieved and by comparing the simulation outcome to results of other research efforts. Although measured performance of existing networks produces optimal results, the simulation model will allow quick network reconfiguration to analyze the network response in various circumstances. Thus, a proposed network configuration may be analyzed before implementation to determine the expected performance of the system.

Finally, the stated objectives of this research have been reached. As always, additional features could be added to this model to produce improvements. For example, future expansion of this model could be provided to link multiple Ethernet cables and provide performance analysis for that configuration. A generalized program to accomodate all configurations --one including random device transmission rather than the specific operations (screen refresh, tape file dumps, and operator requests) --would certainly be a challenge. These improvements would increase the flexibility and usefulness of the model for a wider range of configurations. The current model provides an accurate and valid performance analysis for the system configuration stated.

7.0 REFERENCES AND BIBLIOGRAPHY

[ABA77]
Agrawala, A. K., R. M. Bryant, and J. Agre, ´Analysis of an Ethernet-Like Protocol´, Computer Networking Symposium, December 1977, page 104-111.

[FKP85]
Fritz, James S., Charles F. Kaldenbach, and Louis M. Progar, Local Area Networks: Selection Guidelines, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1985.

[HE82]
Heyman, D. P., ´An Analysis of the Carrier-Sense Multiple Access Protocol´, The Bell System Technical Journal, October 1982, Vol. 61, No. 8, page 2023-2051.

[IN86]
Ingels, F. M., ´System Analysis for the Huntsville Operation Support Center Distributed Computer System´, Contract Final Report 1986, NAS8-34906.

[KI86]
Killen, Harold B., Telecommunications and Data Communication System Design with Troubleshooting, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1985.

[MB76]
Metcalfe, R. M. and D. R. Boggs, ´Ethernet: Distributed Packet Switching for Local Computer Networks´, ACM Communications, July 1976, Vol. 19. No. 7, page 395-404.

[MI85]
Massey, D. and F. M. Ingels, ´System Analysis for the Huntsville Operational Support Center Distributed Computing System´, Annual Report 1985, NAS8-34906.

[SA85]
Stuck, B. W. and E. Arthurs, A Computer and Communications Network Performance Analysis Primer, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1985.

[SH80]
Shoch, John F. and Jon A. Hupp, ´Measured Performance of an Ethernet Local Network´, Communications of the ACM, December 1980, Vol. 23, No. 12, page 711-721.

[SM83]
Sauer, Charles H. and Edward A. MacNair, Simulation of Computer Communication Systems, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1983.

[ST84]
Stallings, William, Local Networks: An Introduction, Macmillan Publishing Company, New York, New York, 1984.

APPENDIX I.

SIMULATION SOURCE LISTING

```
(*****************************************************************************
 * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 *
 *  TITLE:  ETHERNET SIMULATION FOR THE HUNTSVILLE OPERATIONS SUPPORT CENTER
 *
 *
 *  AUTHOR: TERESA BENNETT
 *
 *
 *  DATE:   FEB 23, 1987
 *
 *
 *  PURPOSE: PROVIDE A SIMULATION OF THE HUNTSVILLE OPERATIONS SUPPORT CENTER
 *           ETHERNET NETWORKING SYSTEM CONFIGURATION. THIS ANALYSIS SHALL
 *           RESULT IN SPECIFIC RECOMMENDATIONS FOR IMPROVING SYSTEM
 *           RESPONSIVENESS AND THROUGHPUT CAPACTIY BY (1) IDENTIFYING OPTIMUM
 *           SIZES FOR THE VARIOUS DATA BLOCKS TRANSMITTED VIA THE ETHERNET;
 *           (2) IDENTIFYING PLACES WHERE INTENTIONAL DELAYS SHOULD BE INSERTED
 *           TO ENSURE OPPORTUNITIES FOR SPECIFIC COMMUNICATION VIA THE
 *           ETHERNET; (3) IDENTIFYING OTHER "TUNEABLE" FACTORS WITHIN THE
 *           NETWORK.
 * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 *****************************************************************************)


PROGRAM ETHER (INPUT, OUTPUT, DFILE(LFN=17), AUXOUT (LFN=16), OUT (LFN=18),
       SF1 (LFN=19), SF2 (LFN=20));

(* DFILE - CONTAINS THE RUN TIME CONFIGURATION FOR THE SIMULATION          *)

(* AUXOUT - CONTAINS THE RUN RESULTS: TABLES, PARAMETERS, ETC.             *)

(* OUT - CONTAINS OUTPUT OF INTERNAL VARIABLES DURING THE RUN              *)

(* SF1 - CONTAINS SUMMARY PAGE 1 OF ALL RUNS *)

(* SF2 - CONTAINS SUMMARY PAGE 2 OF ALL RUNS *)

CONST
  DEAD = 9.6E-6;      (* TIME BUS MUST BE QUIET BEFORE NEXT TX CAN OCCUR *)
  MAXNOTERMS = 20;     (* MAXIMUN # OF TERMINALS AVAILABLE PER HOST *)
  MAXNOHOSTS = 100;    (* MAXIMUM # OF HOSTS AVAILABLE ON BUS *)
  MAXNODIST = 10;      (* MAXIMUM # OF DISTRIBUTORS (MAIN FRAMES) *)
  MAXNO = 110;         (* SUM OF MAXNOHOSTS AND MAXNODIS *)
  WAITOFFSET = 1.5E-5; (* USED SO A DISTRIBUTOR WON'T BE SET UP TO SEND REFRESH
                         AND DUMP AT THE SAME TIME *)

TYPE
  ARAY =  PACKED ARRAY [1..17] OF CHAR;
  ARA1 =  PACKED ARRAY [1..40] OF CHAR;
```

```
    TOTBITS: REAL; (* TOTAL BITS OFFERED TO BUS *)
    OFFLOAD: REAL; (* OFFERED LOAD (BITS/SEC)*)
    EFFICIENCY : REAL;
    TP : REAL; (* THEROETICAL THROUGHPUT *)
    SIMTHRUPUT: REAL; (* SIMULATED THROUGHPUT *)
    SIMTIME: REAL; (* SIMULATION BUS RUN TIME*)
    TOTCOLS: INTEGER; (* TOTAL TIMES THERE WAS A BUS COLLISION *)
    NOHOSTS: INTEGER; (* NUMBER OF HOSTS NOT INCLUDING DISTRIBUTORS*)
    NODIST: INTEGER; (* NUMBER OF DISTRIBUTORS *)
    DIS: ARRAY [1..MAXNODIST] OF DISTRECORD;
    HOST: ARRAY [1..MAXNOHOSTS] OF HOSTRECORD;
    TMIT1: TERMRECORD; (* TRANSMITTER DATA *)
    DFILE, AUXOUT, OUT, SF1, SF2: TEXT; (* FILE DECLARATION *)
    DAT: PACKED ARRAY[1..5] OF ARAY;
    ISTAT: INTEGER; (* STATUS RETURNED WHEN BINDING FILES TOGETHER *)
    CURTIME: REAL; (*TIME ON BUS NOW*)
    EBUSRATE: REAL; (* ETHERNET BUS IO RATE*)
    MEANREF: REAL;     (*NO SECONDS BETWEEN REFRESHES TO A HOST*)
    MEANDUMP: REAL;     (*NO SECONDS BETWEEN DUMPS TO A HOST*)
    MEANREQ: REAL;     (*MEAN TIME  BETWEEN OP REQUESTS *)
    GLIY : INTEGER; (*SEED FOR RANDOM NUMBER GENERATOR *)
    GLIR: ARRAY [1..97] OF INTEGER;
    IDUM, RANNUM : INTEGER;
    PDELAY: ARRAY [1..MAXNO,1..MAXNO] OF REAL; (*PROP DELAY BETWEEN EACH *)
    SNOTX : ARRAY [1..MAXNODIST,1..MAXNOHOSTS] OF INTEGER;
    OPREQ: ARRAY [1..MAXNODIST] OF INTEGER; (* OPERATOR REQUEST IS CURRENTLY*)
    FIN : ARRAY [1..MAXNODIST,1..MAXNOHOSTS] OF INTEGER;
    SREFDUMP : ARRAY[1..MAXNODIST,1..MAXNOHOSTS] OF INTEGER;
    HTX : ARRAY[1..MAXNOHOSTS] OF INTEGER;
    SP : INTEGER; (* TOP OF STACK INDICATOR *)
    TEMP : REAL;
    JUNK: INTEGER;
    HCYC : ARRAY [1..MAXNOHOSTS,1..MAXNOTERMS] OF INTEGER;
    ACOLL : INTEGER; (* A COLLISION OCCURRANCE FLAG *)
    SL : INTEGER; (* OPERATOR SELECTION INFORMATION *)
    FN : INTEGER; (* OPERATOR FILE SELECTION *)
    DESCRIPT: ARRAY [1..5] OF ARA1;
    OPCHAR : CHAR;

(******************************************************************************)


 PROCEDURE INITIALIZE;

(******************************************************************************)

VAR
        I, J, K: INTEGER;

BEGIN
        DAT[1] := 'DATA1            ';
```

```
   STRTYP = PACKED ARRAY [1..40] OF CHAR;


   TERMRECORD = RECORD
                 TNUM: INTEGER; (*TERMINAL NUMBER - TERMINAL IDENTIFICATION*)
                 HNUM: INTEGER; (* DEVICE NUMBER - HOST IDENTIFICATION*)
                 TXTIME: REAL; (* ARRIVAL TIME FOR PKT TX *)
                 NUMCOLS: INTEGER; (* NUMBER OF TIMES COLLISION REPEATED - USED
                                   TO CALCULATE EXP BACKOFF TIMES *)
                 RCVR: INTEGER; (* HOST TO RECEIVE TRANSMISSION *)
        END;


   HOSTRECORD = RECORD
                 TERM: ARRAY [1..MAXNO] OF TERMRECORD;
                 NOTERMS: INTEGER; (* NUMBER OF TERMINALS / HOST*)
                 IORATE: REAL; (* I/O RATE OF HOST - TIME TO FILL HOST BUFFER *)
                 BITPKT: REAL; (* NUM OF BITS/PKT*)
                 DISTANCE: REAL; (*DISTANCE FROM PT A FOR HOST *)
                 SLTIME: REAL; (* SLOT TIME = BITPKT/ETHERRATE*)
                 COLTIME: REAL; (* COLISION WAITING TIME *)
                 WTTIME: REAL; (* DEFER WAITING TIME *)
                 NOCOLS: INTEGER; (* NUM OF COLISIONS FOR HOST*)
                 NOWAITS: INTEGER; (* NUM OF WAITS FOR HOST *)
                 MINWAIT: REAL; (* MIN WAIT TIME FOR ANY PKT *)
                 MAXWAIT: REAL; (* MAX WAITING TIME FOR ANY PKT TX *)
                 MAXRCVRWT: INTEGER; (* RECEIVER THAT HAD TO WAIT LONGEST*)
                 PKTSTX : INTEGER; (* NUM OF PACKETS TX BY HOST *)
                 PKTSRX : INTEGER; (* NUM OF PACKETS RX BY HOST *)
        END;


   DISTRECORD = RECORD
                 DIST: ARRAY [1..2] OF HOSTRECORD;
                 REFTIME: REAL; (* TIME TO REFRESH HOST *)
                 REFHST: INTEGER; (* NEXT HOST TO REFRESH *)
                 DUMPTIME: REAL; (* TIME TO DUMP TO HOST *)
                 DUMPHST: INTEGER; (* HOST TO DUMP TO *)
                 PKTSREF: INTEGER; (* SPLIT - NO TX TO REF HOST*)
                 PKTSDUMP: INTEGER; (* SPLIT - NO TX TO DUMP TO HOST*)
                 NOTX: INTEGER; (* NO PKTS LEFT TO TX FOR A DUMP OR REFRESH *)
                 REFDUMP: INTEGER; (* DOING REF=1 OR DOING DUMP=2 *)
        END;

VAR

    CLOCK: REAL; (* CURRENT TIME ON BUS *)
    BUSBUSY: REAL; (* TIME BUS BUSY - TRANSMITTING & WAITING*)
    USAGE: REAL; (* TIME BUS IN USE - TRANSMITTING *)
    IDLE: REAL; (* TIME BUS IDLE *)
    AVGBB, AVGUS, AVGID: REAL; (* AVERAGES *)
    TOTPKTSTX: INTEGER; (* TOTAL PKTS SUCCESSFULLY TX *)
```

```
DAT[2] := 'DATA2            ';
DAT[3] := 'DATA3            ';
DAT[4] := 'DATA4            ';
DAT[5] := 'DATA5            ';

SP := 0;
FOR I := 1 TO MAXNODIST DO
   OPREQ[I] := 0;
FOR I := 1 TO MAXNOHOSTS DO
   HTX[I] := 0;
FOR I := 1 TO MAXNOHOSTS DO
  FOR J := 1 TO MAXNODIST DO
     FIN[J,I] := 0;

FOR I := 1 TO MAXNOHOSTS DO
        WITH HOST[I] DO
        BEGIN
                NOTERMS := 0;
                IORATE := 0.0;
                BITPKT := 0.0;
                DISTANCE := 0.0;
                SLTIME := 0.0;
                COLTIME := 0.0;
                WTTIME := 0.0;
                NOCOLS := 0;
                NOWAITS := 0;
                MINWAIT := 999.9;
                MAXWAIT := 0.0;
                MAXRCVRWT := 0;
                PKTSTX := 0;
                PKTSRX := 0;
                FOR J := 1 TO MAXNOTERMS DO
                        WITH TERM[J] DO
                        BEGIN
                                TNUM := 0;
                                HNUM := 0;
                                TXTIME := 0.0;
                                NUMCOLS := 0;
                                RCVR := 0;
                        END;
        END;
FOR K:= 1 TO MAXNODIST DO
        WITH DIS[K] DO
        BEGIN
                REFTIME := 0.0;
                REFHST := 0;
                DUMPTIME := 0.0;
                DUMPHST := 0;
                PKTSREF := 0;
                PKTSDUMP := 0;
                NOTX := 0;
                REFDUMP := 0;
        FOR J := 1 TO 2 DO
```

```
           WITH DIST[J] DO
           BEGIN
                   NOTERMS := 0;
                   IORATE := 0.0;
                   BITPKT := 0.0;
                   DISTANCE := 0.0;
                   SLTIME := 0.0;
                   COLTIME := 0.0;
                   WTTIME := 0.0;
                   NOCOLS := 0;
                   NOWAITS := 0;
                   MINWAIT := 999.9;
                   MAXWAIT := 0.0;
                   MAXRCVRWT := 0;
                   PKTSTX := 0;
                   PKTSRX := 0;
                   WITH TERM[1] DO
                           BEGIN
                                   TNUM := 0;
                                   HNUM := 0;
                                   TXTIME := 0.0;
                                   NUMCOLS := 0;
                                   RCVR := 0;
                           END;
           END;
   END;
   CLOCK := 0.0;
   BUSBUSY := 0.0;
   USAGE := 0.0;
   IDLE := 0.0;
   TOTPKTSTX := 0;
   TOTBITS := 0.0;
   NOHOSTS := 0;
   TOTCOLS := 0;
   EBUSRATE := 0.0;
   MEANREF := 0.0;
   MEANDUMP := 0.0;
   MEANREQ := 0.0;
   IDUM := 0;

   FOR I:=1 TO MAXNO DO
           FOR J:=1 TO MAXNO DO
                   PDELAY [I,J] := 0.0;

   FOR I:= 1 TO MAXNOHOSTS DO
           FOR J:= 1 TO MAXNOTERMS DO
                  HCYC[I,J] := 1;
END;

(*****************************************************************************)
```

```
 FUNCTION RAN (VAR IDUM: INTEGER): REAL;

(***************************************************************************)
   CONST
        M = 714025;
        IA = 1366;
        IC = 150889;
        RM = 1.400512E-6;

   VAR
        J : INTEGER;


    BEGIN
        IF (IDUM < 0) THEN
           BEGIN
                IDUM := (IC-IDUM) MOD M;
                FOR J := 1 TO 97 DO
                        BEGIN
                        IDUM := (IA*(IDUM MOD 6030) + IC) MOD M;
                        GLIR[J] := IDUM;
                        END;
                IDUM := ((IA*(IDUM MOD 6030))+IC) MOD M;
                GLIY := IDUM;
           END;
        J := 1+(97*(GLIY MOD 86480)) DIV M;
        IF (J > 97) OR (J < 1) THEN
                WRITELN ('SOMETHING IS WRONG WITH THE RANDON NUMBER GENERATOR');
        GLIY := GLIR[J];
        RAN := GLIY • RM;
        IDUM := ((IA*(IDUM MOD 6030))+IC) MOD M;
        GLIR[J] := IDUM;
    END;

(***************************************************************************)


 PROCEDURE GETDATA;

(***************************************************************************)

VAR

   I, J, K : INTEGER;

BEGIN
```

```
(* DFILE - OPERATOR SELECTED FILE. GO READ SIMULATION INPUT DATA *)

        RESET(DFILE);
        READLN(DFILE,DESCRIPT[FN]);
        READLN (DFILE,IDUM);
        RANNUM := IDUM;
        IDUM := 0 - IDUM;
        READLN(DFILE,SIMTIME);
        READLN (DFILE,EBUSRATE);
        READLN (DFILE,NODIST);
        READLN (DFILE,NOHOSTS);
        READLN (DFILE,MEANREF);
        READLN (DFILE,MEANDUMP);
        READLN (DFILE,MEANREQ);
        FOR I := 1 TO NODIST DO
                BEGIN
                        WITH DIS[I] DO
                        BEGIN
                        FOR J:=1 TO 2 DO
                          WITH DIST[J] DO
                          BEGIN
                          READLN (DFILE,NOTERMS);
                          TERM[1].HNUM := NOHOSTS + I;
                          TERM[1].TNUM := 1;
                          READLN (DFILE,IORATE);
                          READLN (DFILE,BITPKT);
                          READLN (DFILE,DISTANCE);
                          SLTIME := BITPKT/ EBUSRATE;
                          END;
                        REFTIME := MEANREF/NOHOSTS;
                        REFHST := 1;
                        DUMPTIME:=MEANDUMP/NOHOSTS;
                        DUMPHST := 1;
                        READLN (DFILE,PKTSREF);
                        READLN (DFILE,PKTSDUMP);
                        REFDUMP := 1;
                        END;
                END;
        FOR I := 1 TO NOHOSTS DO
                BEGIN
                        WITH HOST[I] DO
                        BEGIN
                        READLN (DFILE,NOTERMS);
                        READLN (DFILE,IORATE);
                        READLN (DFILE,BITPKT);
                        READLN (DFILE,DISTANCE);
                        SLTIME := BITPKT/ EBUSRATE;
                        FOR J := 1 TO NOTERMS DO
                                BEGIN
                                        WITH TERM[J] DO
                                        BEGIN
                                        TNUM := J;
                                        HNUM := I;
                                        TXTIME := -(MEANREQ/NOTERMS) *
                                                LN(RAN(IDUM));
                                        RCVR := NOHOSTS + 1;
                                        END;
                                END;
                        FOR J := 1 TO NOTERMS-1 DO
                           FOR K := J+1 TO NOTERMS DO
                             IF TERM[J].TXTIME > TERM[K].TXTIME THEN
                                BEGIN
                                TEMP := TERM[J].TXTIME;
                                TERM[J].TXTIME := TERM[K].TXTIME;
                                TERM[K].TXTIME := TEMP;
                                END;
                            FOR J := 2 TO NOTERMS DO
                            TERM[J].TXTIME := TERM[J].TXTIME + TERM[J-1].TXTIME;
                        END;
                END;
      CLOSE (DFILE);
END;

(*****************************************************************************)


 PROCEDURE DISPLAY;

(*****************************************************************************)

(* DISPLAY SIMULATION CONFIGURATION DATA TO OPERATOR TERMINAL *)

VAR
   INP1, INP2: INTEGER;
   INP3: CHAR;


BEGIN

 REPEAT

   WRITELN(' ');
   WRITELN(' ');
   WRITELN(' ');
   WRITELN('           DISPLAY DATA MENU ');
   WRITELN(' ');
   WRITELN(' ');
   WRITELN('1  -  GENERAL CONFIGURATION INFORMATION (GLOBAL DATA) ');
   WRITELN('2  -  MAIN FRAME PARAMETERS ');
   WRITELN('3  -  HOST PARAMETERS ');
   WRITELN('4  -  RETURN TO PREVIOUS MENU ');
   WRITELN(' ');
   REPEAT
      WRITELN('ENTER SELECTION: ');
      READLN(INP1);
   UNTIL ((INP1 >=1) AND (INP1 <= 4));

(* OPERATOR SELECTED GENERAL CONFIGURATION INFORMATION TO BE DISPLAYED *)

IF INP1 = 1 THEN
   BEGIN
      WRITELN('DESCRIPTION OF CONFIGURATION: ',DESCRIPT[FN]);
      WRITELN(' ');
      WRITELN('RANDOM NUMBER: ',RANNUM);
      WRITELN('SIMULATION RUN TIME: ',SIMTIME);
      WRITELN('ETHERNET BUS I/O RATE: ',EBUSRATE);
      WRITELN(' ');
      WRITELN('NUMBER OF MAIN FRAMES: ',NODIST);
      WRITELN('NUMBER OF HOSTS: ',NOHOSTS);
      WRITELN(' ');
      WRITELN('REFRESHES TO HOSTS IN SECONDS: ',MEANREF);
      WRITELN('DUMPS TO HOSTS IN SECONDS: ',MEANDUMP);
      WRITELN('OPERATOR REFRESH REQUESTS IN SECONDS: ',MEANREQ);
      WRITELN(' ');
      WRITELN('ENTER <RETURN> TO CONTINUE ');
      WHILE NOT EOLN DO
          READ(INP3);
   END;

(* OPERATOR SELECTED MAIN FRAME DATA TO BE DISPLAYED *)

 IF INP1 = 2 THEN
  BEGIN
   REPEAT
     WRITELN('ENTER MAIN FRAME NUMBER (1 TO ',NODIST:2,')');
     READLN(INP2);
   UNTIL ((INP2 >= 1) AND (INP2 <= NODIST));

   WRITELN('MAIN FRAME NUMBER: ',INP2);
   WRITELN(' ');
   WRITELN('I/O RATE OF MAIN FRAME: ',DIS[INP2].DIST[1].IORATE);
   WRITELN('DISTANCE FROM REFRENCE POINT: ',DIS[INP2].DIST[1].DISTANCE);
   WRITELN(' ');
   WRITELN('REFRESH OPERATION:');
   WRITELN('     BITS/PACKET: ',DIS[INP2].DIST[1].BITPKT);
   WRITELN('     SLTIME: ',DIS[INP2].DIST[1].SLTIME);
   WRITELN('     NUMBER OF PACKETS SENT TO EACH TERMINAL ',DIS[INP2].PKTSREF);
   WRITELN(' ');
   WRITELN('DUMP OPERATION:');
   WRITELN('     BITS/PACKET: ',DIS[INP2].DIST[2].BITPKT);
   WRITELN('     SLTIME: ',DIS[INP2].DIST[2].SLTIME);
   WRITELN('     NUMBER OF PACKETS SENT TO EACH TERMINAL ',DIS[INP2].PKTSDUMP);
   WRITELN(' ');
   WRITELN('ENTER <RETURN> TO CONTINUE ');
   WHILE NOT EOLN DO
```

```
       READ (INP3);
  END;

(* OPERATOR SELECTED HOST INFORMATION TO BE DISPLAYED *)

IF INP1 = 3 THEN
  BEGIN
   REPEAT
     WRITELN('ENTER HOST NUMBER (1 TO ',NOHOSTS:2,')');
     READLN(INP2);
   UNTIL ((INP2 >= 1) AND (INP2 <= NOHOSTS));

   WRITELN('HOST NUMBER: ',INP2);
   WRITELN(' ');
   WRITELN('NUMBER OF TERMINALS FOR HOST: ',HOST[INP2].NOTERMS);
   WRITELN('I/O RATE OF HOST: ',HOST[INP2].IORATE);
   WRITELN('BITS PER PACKET: ',HOST[INP2].BITPKT);
   WRITELN('DISTANCE FROM REFRENCE POINT: ',HOST[INP2].DISTANCE);
   WRITELN('SLOT TIME: ',HOST[INP2].SLTIME);
   WRITELN(' ');
   WRITELN('ENTER <RETURN> TO CONTINUE ');
   WHILE NOT EOLN DO
      READ(INP3);
  END;

(* OPERATOR IS READY TO GO BACK TO MAIN MENU *)

UNTIL ( INP1 = 4);
END;


(******************************************************************************)


 PROCEDURE REALGETOP(VIEW:STRTYP; VAR VALUE: REAL);


(******************************************************************************)

(* QUERY OPERATOR FOR A REAL NUMBER TO REPLACE VARIABLE DISPLAYED.
   VIEW CONTAINS THE VARIABLE TO BE DISPLAYED ON THE OP TERMINAL.
   VALUE WILL BE RETURNED WITH DATA PASSED OR WHAT OP ENTERS.   *)


VAR
   INP1: REAL;

BEGIN

   INP1 := 99999.999;
   WRITELN('PRESS <RETURN> TO RETAIN CURRENT VALUE OR ');
   WRITELN('ENTER ',VIEW,' [CURRENT: ',VALUE,']');

   WHILE NOT EOLN DO
      READ (INP1);
   READLN;

   IF INP1 <> 99999.999 THEN
        VALUE := INP1;

   WRITELN(' ');

END;

(*****************************************************************************)


 PROCEDURE INTGETOP(VIEW:STRTYP; VAR VALUE: INTEGER);

(*****************************************************************************)

(* SEE REALGETOP - INTGETOP ESTABLISHED FOR INTEGER VALUES    *)


VAR
   INP1: INTEGER;

BEGIN

   INP1 := 99999;
   WRITELN('PRESS <RETURN> TO RETAIN CURRENT VALUE OR ');
   WRITELN('ENTER ',VIEW,' [CURRENT: ',VALUE,']');

   WHILE NOT EOLN DO
      READ (INP1);
   READLN;

   IF INP1 <> 99999 THEN
        VALUE := INP1;

   WRITELN(' ');

END;

(*****************************************************************************)


 PROCEDURE MODIFY;
```

```
(***************************************************************************)

(* ALLOWS OPERATOR TO MODIFY SIMULATION PARAMETERS *)

VAR
   OPDISP: STRTYP;
   INP1: INTEGER;
   INP2: INTEGER;
   INDES: ARRAY [1..5] OF ARA1;
   I,J,K: INTEGER;

BEGIN

 REPEAT

   WRITELN(' ');
   WRITELN(' ');
   WRITELN(' ');
   WRITELN('           MODIFY PARAMETERS MENU ');
   WRITELN(' ');
   WRITELN(' ');
   WRITELN('1  -  GENERAL CONFIGURATION INFORMATION (GLOBAL DATA) ');
   WRITELN('2  -  MAIN FRAME PARAMETERS ');
   WRITELN('3  -  HOST PARAMETERS ');
   WRITELN('4  -  RETURN TO PREVIOUS MENU ');
   WRITELN(' ');

   REPEAT
      WRITELN('ENTER SELECTION: ');
      READLN(INP1);
   UNTIL ((INP1 >=1) AND (INP1 <= 4));

(* OPERATOR SELECTED GENERAL CONFIGURATION DATA TO MODIFY *)

IF INP1 = 1 THEN
   BEGIN
      INDES[FN] := 'AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA';
      WRITELN('PRESS RETURN TO RETAIN CURRENT VALUE OR ');
      WRITELN('ENTER NEW DESCRIPTION, CURRENT: ',DESCRIPT[FN]);
      WHILE NOT EOLN DO
         READ(INDES[FN]);
      READLN;
      IF INDES[FN] <> 'AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA' THEN
         DESCRIPT[FN] := INDES[FN];
      WRITELN(' ');

      OPDISP := 'RANDON NUMBER                          ';
      INTGETOP(OPDISP,RANNUM);
      IDUM := 0 - RANNUM;

      OPDISP := 'SIMULATION RUN TIME                    ';
      REALGETOP(OPDISP,SIMTIME);
      OPDISP := 'ETHERNET BUS I/O RATE                   ';
      REALGETOP(OPDISP,EBUSRATE);

      OPDISP := 'NUMBER OF MAIN FRAMES                   ';
      INTGETOP(OPDISP,NODIST);

      OPDISP := 'NUMBER OF HOSTS                         ';
      INTGETOP(OPDISP,NOHOSTS);

      OPDISP := 'REFRESHES TO HOSTS IN SECONDS           ';
      REALGETOP(OPDISP,MEANREF);

      OPDISP := 'DUMPS TO HOSTS IN SECONDS               ';
      REALGETOP(OPDISP,MEANDUMP);

      OPDISP := 'OPERATOR REFRESH REQUESTS IN SECONDS    ';
      REALGETOP(OPDISP,MEANREQ);
   END;

(* OPERATOR SELECTED TO MODIFY MAIN FRAME DATA *)

 IF INP1 = 2 THEN
  BEGIN
   REPEAT
     WRITELN('ENTER MAIN FRAME NUMBER (1 TO ',NODIST:2,')');
     READLN(INP2);
   UNTIL ((INP2 >= 1) AND (INP2 <= NODIST));

   WITH DIS[INP2] DO
    BEGIN

   WRITELN('MAIN FRAME NUMBER: ',INP2);
   WRITELN(' ');

   OPDISP := 'I/O RATE OF MAIN FRAME                  ';
   REALGETOP(OPDISP,DIST[1].IORATE);
   IF DIST[1].IORATE <> DIST[2].IORATE THEN
       DIST[2].IORATE := DIST[1].IORATE;

   OPDISP := 'DISTANCE FROM REFERENCE POINT           ';
   REALGETOP(OPDISP,DIST[1].DISTANCE);
   IF DIST[1].DISTANCE <> DIST[2].DISTANCE THEN
       DIST[2].DISTANCE := DIST[1].DISTANCE;

   WRITELN(' ');
   WRITELN('REFRESH OPERATION:');
   WRITELN(' ');

   OPDISP := 'BITS/PACKET                             ';
   REALGETOP(OPDISP,DIST[1].BITPKT);

   OPDISP := 'NUMBER OF PACKETS SET TO EACH TERMINAL  ';
   INTGETOP(OPDISP,PKTSREF);

   WRITELN(' ');
   WRITELN('DUMP OPERATION:');
   WRITELN(' ');

   OPDISP := 'BITS/PACKET                             ';
   REALGETOP(OPDISP,DIST[2].BITPKT);

   OPDISP := 'NUMBER OF PACKETS SENT TO EACH TERMINAL ';
   INTGETOP(OPDISP,PKTSDUMP);
   WRITELN(' ');

   END;
  END;

(* OPERATOR SELECTED TO MODIFY HOST DATA *)

IF INP1 = 3 THEN
  BEGIN
   REPEAT
     WRITELN('ENTER HOST NUMBER (1 TO ',NOHOSTS:2,')');
     READLN(INP2);
   UNTIL ((INP2 >= 1) AND (INP2 <= NOHOSTS));

  WITH HOST[INP2] DO
   BEGIN

   WRITELN(' ');
   WRITELN('HOST NUMBER: ',INP2);
   WRITELN(' ');
   OPDISP := 'NUMBER OF TERMINALS FOR HOST            ';
   INTGETOP(OPDISP,NOTERMS);

   OPDISP := 'I/O RATE OF HOST                        ';
   REALGETOP(OPDISP,IORATE);

   OPDISP := 'BITS PER PACKET                         ';
   REALGETOP(OPDISP,BITPKT);

   OPDISP := 'DISTANCE FROM REFERENCE POINT           ';
   REALGETOP(OPDISP,DISTANCE);

   WRITELN(' ');
  END;
  END;

(* RETURN TO PREVIOUS MENU WAS SELECTED *)
UNTIL ( INP1 = 4);

(* UPDATE VARIABLES WHICH MAY BE AFFECTED BY OPERATOR MODIFICATION *)

     FOR I := 1 TO NODIST DO
```

```
       WITH DISC[I] DO
         BEGIN
                 DIST[1].SLTIME := DIST[1].BITPKT / EBUSRATE;
                 DIST[2].SLTIME := DIST[2].BITPKT / EBUSRATE;
                 REFTIME := MEANREF/NOHOSTS;
                 REFHST := 1;
                 DUMPTIME:=MEANDUMP/NOHOSTS;
                 DUMPHST := 1;
                 DIST[1].NOTERMS := 1;
                 DIST[1].TERM[1].HNUM := NOHOSTS + I;
                 DIST[1].TERM[1].TNUM := 1;
                 DIST[2].NOTERMS := 1;
                 DIST[2].TERM[1].HNUM := NOHOSTS + I;
                 DIST[2].TERM[1].TNUM := 1;
                 REFDUMP := 1;
       END;

          FOR I := 1 TO NOHOSTS DO
             WITH HOST[I] DO
                BEGIN
                 SLTIME := BITPKT / EBUSRATE;
                  FOR J := 1 TO NOTERMS DO
                                BEGIN
                                        WITH TERM[J] DO
                                        BEGIN
                                        TNUM := J;
                                        HNUM := I;
                                        TXTIME := -(MEANREQ/NOTERMS) *
                                                LN(RAN(IDUM));
                                        RCVR := NOHOSTS + 1;
                                        END;
                                END;
                        FOR J := 1 TO NOTERMS-1 DO
                           FOR K := J+1 TO NOTERMS DO
                             IF TERM[J].TXTIME > TERM[K].TXTIME THEN
                                BEGIN
                                TEMP := TERM[J].TXTIME;
                                TERM[J].TXTIME := TERM[K].TXTIME;
                                TERM[K].TXTIME := TEMP;
                                END;
                            FOR J := 2 TO NOTERMS DO
                            TERM[J].TXTIME := TERM[J].TXTIME + TERM[J-1].TXTIME;
                        END;

END;


(*****************************************************************************)
```

```
 PROCEDURE STORE;

(*****************************************************************************)

(* WRITE SIMULATION RUN PARAMETERS TO DATA FILE SELECTED BY OPERATOR *)


VAR

   I, J, K : INTEGER;


BEGIN

        REWRITE(DFILE);
        WRITELN(DFILE,DESCRIPT[FN]);
        WRITELN (DFILE,RANNUM);
        WRITELN(DFILE,SIMTIME);
        WRITELN (DFILE,EBUSRATE);
        WRITELN (DFILE,NODIST);
        WRITELN (DFILE,NOHOSTS);
        WRITELN (DFILE,MEANREF);
        WRITELN (DFILE,MEANDUMP);
        WRITELN (DFILE,MEANREQ);
        FOR I := 1 TO NODIST DO
                BEGIN
                        WITH DIS[I] DO
                        BEGIN
                        FOR J:=1 TO 2 DO
                          WITH DIST[J] DO
                          BEGIN
                          WRITELN (DFILE,NOTERMS);
                          WRITELN (DFILE,IORATE);
                          WRITELN (DFILE,BITPKT);
                          WRITELN (DFILE,DISTANCE);
                          END;
                        WRITELN (DFILE,PKTSREF);
                        WRITELN (DFILE,PKTSDUMP);
                        END;
                END;
        FOR I := 1 TO NOHOSTS DO
                BEGIN
                        WITH HOST[I] DO
                        BEGIN
                        WRITELN (DFILE,NOTERMS);
                        WRITELN (DFILE,IORATE);
                        WRITELN (DFILE,BITPKT);
                        WRITELN (DFILE,DISTANCE);
                        END;
                END;
END;
```

```
(******************************************************************************)

 PROCEDURE CONFIGURE;

(******************************************************************************)

(* THIS ROUTINE WILL QUERY THE OPERATOR FOR THE DATA FILE OF HIS CHOICE
   BY DISPLAYING THE DESCRIPTION FROM WITHIN THE FILE IN A MENU FORMAT *)

    VAR
        I, J, K : INTEGER;

    BEGIN
FOR I:= 1 TO 5 DO
  BEGIN
     BIND(DFILE,DAT[I],ISTAT);
     READLN(DFILE,DESCRIPT[I]);
     CLOSE(DFILE);
  END;

WRITELN('THERE ARE 5 DATA FILES CURRENTLY AVAILABLE FOR OPERATOR USE. ');
WRITELN('EACH FILE CONTAINS CONFIGURATION INFORMATION FROM PREVIOUS RUNS. ');
WRITELN('THE PARAMETERS MAY BE DISPLAYED AND MODIFIED BEFORE SIMULATION.');
WRITELN('SELECT THE NUMBER BELOW DESCRIBING THE FILE DESIRED. ');
WRITELN(' ');

FOR I:= 1 TO 5 DO
    WRITELN('        ',I:1,' - ',DESCRIPT[I]);

WRITELN(' ');
WRITELN(' ');
FN := 0;
REPEAT
   WRITELN('ENTER SELECTION: ');
   READLN(FN);
UNTIL ((FN >= 1) AND (FN <= 5));

BIND(DFILE,DAT[FN],ISTAT);

GETDATA;

REPEAT

WRITELN(' ');
WRITELN(' ');
WRITELN(' ');
WRITELN('           ETHERNET SIMULATION MENU ');
WRITELN(' ');
WRITELN(' ');
WRITELN('1  -  DISPLAY CONFIGURATION INFORMATION ON TERMINAL ');
WRITELN('2  -  MODIFY CONFIGURATION INFORMATION      ');
WRITELN('3  -  STORE MODIFIED INFORMATION TO FILE SELECTED ');
WRITELN('4  -  RUN SIMULATION                             ');
WRITELN('5  -  EXIT PROGRAM                               ');
WRITELN(' ');
SL := 0;
REPEAT
    WRITELN('ENTER SELECTION:');
    READLN(SL);
UNTIL ((SL >= 1) AND (SL <= 5));

IF SL = 1 THEN DISPLAY;
IF SL = 2 THEN MODIFY;
IF SL = 3 THEN
   BEGIN
        BIND(DFILE,DAT[FN],ISTAT);
        STORE;
   END;

UNTIL ((SL = 4) OR (SL = 5));

IF SL = 4 THEN
   BEGIN
        FOR I:=1 TO NOHOSTS DO
                BEGIN
                FOR J:=1 TO NOHOSTS DO
                        IF I<>J THEN
                                PDELAY[I,J] := (ABS(HOST[I].DISTANCE-HOST[J].
                                         DISTANCE)) • 1.27E-9
                        ELSE
                                PDELAY [I,J] := 0.0;
                FOR J:= 1 TO NODIST DO
                PDELAY[I,NOHOSTS+J]:=(ABS(HOST[I].DISTANCE-DIS[J].
                        DIST[1].DISTANCE)) • 1.27E-9;
                END;
        FOR I:=1 TO NODIST DO
                FOR J:=1 TO NOHOSTS DO
                        PDELAY[NOHOSTS+I,J] := (ABS(DIS[I].DIST[1].
                                 DISTANCE-HOST[J].DISTANCE))*1.27E-9;

        FOR I := 1 TO NODIST DO
                FOR J:= 1 TO NODIST DO
                        BEGIN
                        PDELAY[NOHOSTS+I,NOHOSTS+J] := (ABS(DIS[I].
                                DIST[1].DISTANCE-DIS[J].DIST[1].
                                DISTANCE)) • 1.27E-9;
                        END;
     END;
   END;
```

C -2

```
(****************************************************************************)


PROCEDURE PRINTCONFIG;


(****************************************************************************)

(* WRITE SIMUALATION PARAMETERS TO AUXOUT *)

    VAR
        I, J, K : INTEGER;

    BEGIN
        REWRITE(AUXOUT);
        REWRITE (OUT);
        WRITELN (AUXOUT,'          ETHERNET SIMULATION PARAMETERS ');
        WRITELN (AUXOUT,' ');
        WRITELN (AUXOUT,' ');
        WRITELN (AUXOUT,' ');
        WRITELN (AUXOUT,'ETHERNET BUS IO RATE: ',EBUSRATE);
        WRITELN (AUXOUT,' ');
        WRITELN (AUXOUT,'NUMBER OF MAIN FRAMES: ',NODIST);
        WRITELN (AUXOUT,'NUMBER OF HOSTS: ',NOHOSTS);
        WRITELN (AUXOUT,' ');
        WRITELN (AUXOUT,'REFRESHES TO HOSTS IN SECONDS: ',MEANREF);
        WRITELN (AUXOUT,'DUMPS TO HOSTS IN SECONDS: ',MEANDUMP);
        WRITELN (AUXOUT,'OPERATOR REFRESH REQUESTS IN SECONDS: ',MEANREQ);
        WRITELN (AUXOUT, ' ');
        WRITELN (AUXOUT, ' ');
        FOR I := 1 TO NODIST DO
            WITH DIS[I] DO
              BEGIN
                  WRITELN(AUXOUT,'MAIN FRAME NUMBER: ',I);
                  WRITELN(AUXOUT,' ');
                  WRITELN(AUXOUT,'I/O RATE OF MAIN FRAME: ',DIST[1].IORATE);
                  WRITELN(AUXOUT,'DISTANCE FROM REFRENCE POINT: ',
                                DIST[1].DISTANCE);
                  WRITELN(AUXOUT,'FIRST REFRESH TIME: ',REFTIME,
                                 ' TO HOST: ',REFHST:2);
                  WRITELN(AUXOUT,'FIRST DUMP TIME: ',DUMPTIME,
                                 ' TO HOST: ',DUMPHST:2);
                  WRITELN(AUXOUT,' ');
                  WRITELN(AUXOUT,'REFRESH OPERATION: ');
                  WRITELN(AUXOUT,'     BITS/PACKET: ',DIST[1].BITPKT);
                  WRITELN(AUXOUT,'     SLTIME: ',DIST[1].SLTIME);
                  WRITELN(AUXOUT,'     NUMBER OF PACKETS SENT TO EACH ',
                                'TERMINAL: ',PKTSREF);
                  WRITELN(AUXOUT,' ');
                  WRITELN(AUXOUT,'DUMP OPERATION: ');
                  WRITELN(AUXOUT,'     BITS/PACKET: ',DIST[2].BITPKT);
                  WRITELN(AUXOUT,'      SLTIME: ',DIST[2].SLTIME);
                  WRITELN(AUXOUT,'      NUMBER OF PACKETS SENT TO EACH ',
                                 'TERMINAL: ',PKTSDUMP);
                  WRITELN(AUXOUT,' ');
                END;

        FOR I := 1 TO NOHOSTS DO
                BEGIN
                        WITH HOST[I] DO
                        BEGIN
                        WRITELN (AUXOUT,' ');
                        WRITELN (AUXOUT,'HOST NUMBER: ',I);
                        WRITELN (AUXOUT,' ');
                        WRITELN (AUXOUT,'NUMBER OF TERMINALS: ',NOTERMS);
                        WRITELN (AUXOUT,'I/O RATE OF HOST: ',IORATE);
                        WRITELN (AUXOUT,'BITS PER PACKET: ',BITPKT);
                        WRITELN (AUXOUT,'DISTANCE FROM REFERENCE POINT: ',
                                DISTANCE);
                        WRITELN (AUXOUT, 'SLOT TIME: ',SLTIME);
(*                      IF NOTERMS <> 0 THEN
                        FOR J := 1 TO NOTERMS DO
                                BEGIN
                                        WITH TERM[J] DO
                                        WRITELN (AUXOUT,'TXTIME FOR TERM ',J:2,
                                                ': ',TXTIME);
                                END;                                      *)
                        END;
                        WRITELN (AUXOUT,' ');
                 END;

WRITELN(AUXOUT,' ');
WRITELN(AUXOUT,' ');
WRITELN(AUXOUT,' ');

    END;

(******************************************************************************)


 PROCEDURE FINDNEXT (VAR TMIT1 : TERMRECORD);


(******************************************************************************)

(* FINDS THE NEXT MAINFRAME OR HOST WITH THE SMALLEST TRAMSMIT TIME.
   AND IS SELECTED TO BE THE TRANSMITTER FOR THIS ITERATION  *)

    VAR
        I,J,K : INTEGER;
        FLAG: INTEGER;
        TSLTIME : REAL;
```

```
BEGIN
    TMIT1.TXTIME := 1000.0;

    (* FIND DISTRIBUTOR WITH NEXT SMALLEST TX TIME *)
    FOR I:= 1 TO NODIST DO
    BEGIN
        (* WHEN NOTX=0 THEN THE DISTRIBUTOR IS READY TO SERVICE THE    *)
        (* NEXT SCREEN REFRESH OR TAPE FILE DUMP, OTHERWISE THAT       *)
        (* DISTRIBUTOR HAS PACKETS LEFT TO BE TRANSMITTED              *)
         IF DIS[I].NOTX = 0 THEN
            BEGIN
            (* FOR A PARTICULAR DISTRIBUTOR FIND THE SMALLEST          *)
            (* TIME BETWEEN THE SREEN REFRESH OR THE TAPE FILE DUMP    *)
            IF DIS[I].REFTIME < TMIT1.TXTIME THEN
               BEGIN
                    TMIT1.TXTIME := DIS[I].REFTIME;
                    TMIT1.RCVR := DIS[I].REFHST;
                    TMIT1.HNUM := I + NOHOSTS;
                    TMIT1.TNUM := 1;
                    TMIT1.NUMCOLS := 0;
                    DIS[I].REFDUMP := 1;
                    TSLTIME := DIS[I].DIST[1].SLTIME;
              END;
            IF DIS[I].DUMPTIME < TMIT1.TXTIME THEN
            BEGIN
                    TMIT1.TXTIME := DIS[I].DUMPTIME;
                    TMIT1.RCVR := DIS[I].DUMPHST;
                    TMIT1.HNUM := I + NOHOSTS;
                    TMIT1.TNUM := 1;
                    TMIT1.NUMCOLS := 0;
                    DIS[I].REFDUMP := 2;
                    TSLTIME := DIS[I].DIST[2].SLTIME;
             END;
            END
            ELSE
            BEGIN
                   (* PACKETS REMAIN TO BE TRANSMITTED FROM SCREEN REF *)
                   (* OR TAPE FILE DUMP BEGUN PREVIOUSLY               *)
                    WITH DIS[I].DIST[DIS[I].REFDUMP].TERM[1] DO
                    IF TXTIME < TMIT1.TXTIME THEN
                    BEGIN
                       TMIT1.TXTIME := TXTIME;
                       TMIT1.HNUM := HNUM;
                       TMIT1.TNUM := 1;
                       TMIT1.RCVR := RCVR;
                       TMIT1.NUMCOLS := NUMCOLS;
                       TSLTIME := DIS[I].DIST[DIS[I].REFDUMP].SLTIME;
                    END;
            END;
    END;

   (* ANY OPERATOR REQUESTS READY TO BE TRANSMITTED? IF SO THEN SET TMIT1*)
```

```
FOR I:= 1 TO NOHOSTS DO
BEGIN
  FLAG := 0;
  FOR K:= 1 TO SP DO
    IF HTX[K] = I THEN
       BEGIN
        FLAG := 1;
       END;

        IF FLAG = 0 THEN
        BEGIN
           FOR J := 1 TO HOST[I].NOTERMS DO
                WITH HOST[I] DO
                IF(TERM[J].TXTIME<TMIT1.TXTIME) THEN
                BEGIN
                        TMIT1.TNUM := J;
                        TMIT1.TXTIME := TERM[J].TXTIME;
                        TMIT1.HNUM := TERM[J].HNUM;
                        TMIT1.RCVR := TERM[J].RCVR;
                        TMIT1.NUMCOLS := TERM[J].NUMCOLS;
                        TSLTIME := SLTIME;
                END;
          END
        ELSE
           FOR J:= 1 TO HOST[I].NOTERMS DO
                WITH HOST[I] DO
                    IF ((TERM[J].TXTIME - TMIT1.TXTIME) <= (TSLTIME +
                    9.6E-6 + PDELAY[TMIT1.HNUM,I])) THEN
                        BEGIN
                          TERM[J].TXTIME := TMIT1.TXTIME + TSLTIME +
                          2*9.6E-6  + PDELAY[TMIT1.HNUM,I];
                        END;

END;

IF(TMIT1.HNUM > NOHOSTS) THEN
        IF (DIS[TMIT1.HNUM-NOHOSTS].NOTX = 0) THEN
        BEGIN
                WITH DIS[TMIT1.HNUM-NOHOSTS] DO
                WITH DIST[DIS[TMIT1.HNUM-NOHOSTS].REFDUMP] DO
                BEGIN
                  TERM[1].TXTIME := TMIT1.TXTIME;
                  IF REFDUMP = 1 THEN
                        BEGIN
                        TERM[1].RCVR := REFHST;
                        NOTX := HOST[TERM[1].RCVR].NOTERMS*PKTSREF;
                        TERM[1].NUMCOLS := 0;
                        END
                  ELSE
                  IF REFDUMP = 2 THEN
                        BEGIN
                        TERM[1].RCVR := DUMPHST;
                        NOTX := HOST[TERM[1].RCVR].NOTERMS*PKTSDUMP;
```

```
                                TERM[1].NUMCOLS := 0;
                                END;
                        END;
                    END;
    END;

(***************************************************************************)


 PROCEDURE COLLISION (VAR TERM1: TERMRECORD;VAR TERM2: TERMRECORD;
                      VAR SUM1: REAL; VAR SUM2: REAL);


(***************************************************************************)

(* USED TO CALCULATE THE NEXT TRANSMIT TIME FOR THE STATIONS WHICH COLLIDED *)

VAR

   X : REAL;
   I : INTEGER;
   LP : INTEGER;

BEGIN

(* USED TO CALCULATE THE NEXT TRANSMIT TIME FOR THE STATIONS WHICH COLLIDED *)
    SUM1 := ABS(TERM1.TXTIME-TERM2.TXTIME) + 9.6E-6;
    SUM1 := SUM1 + 32/EBUSRATE;
    SUM2 := SUM1;

    X := 1;
    LP := TERM1.NUMCOLS;
    IF (TERM1.NUMCOLS <> 0) THEN
     BEGIN
       IF TERM1.NUMCOLS > 10 THEN
          LP := 10;
       FOR I := 1 TO LP DO
          X := X * 2;
     END;
    SUM1 := SUM1 + RAN(IDUM) * X * 51.2E-6;

    X := 1;
    LP := TERM2.NUMCOLS;
    IF (TERM2.NUMCOLS <> 0 ) THEN
     BEGIN
       IF TERM2.NUMCOLS > 10 THEN
          LP := 10;
       FOR I := 1 TO LP DO
          X := X * 2;
    END;
    SUM2 := SUM2 + RAN(IDUM) * X * 51.2E-6;
    TERM1.NUMCOLS := TERM1.NUMCOLS + 1;
    IF ACOLL = 0 THEN
        TERM2.NUMCOLS := TERM2.NUMCOLS + 1;
    IF TERM1.NUMCOLS >= 16 THEN
        TERM1.NUMCOLS := 0;
    IF TERM2.NUMCOLS >= 16 THEN
        TERM2.NUMCOLS := 0;
END;

(**************************************************************************************)


 PROCEDURE COLLIDE (SUM : REAL; VAR CHOST : HOSTRECORD;TNUM : INTEGER);


(**************************************************************************************)


  BEGIN

(* USED TO UPDATE STATION PARAMETERS *)


  WITH CHOST DO
  BEGIN

    NOCOLS := NOCOLS + 1;
    COLTIME := COLTIME + SUM;
    IF (MINWAIT > SUM) THEN MINWAIT := SUM;
    IF (MAXWAIT < SUM) THEN
        BEGIN
        MAXWAIT := SUM;
        MAXRCVRWT := TERM[TNUM].RCVR;
        END;

  END;

END;



(**************************************************************************************)


 PROCEDURE ACOLLISION (VAR THOST: HOSTRECORD);

(**************************************************************************************)
```

```
(* COMPARE TRANSMITTER WITH ALL OTHER STATIONS TO SEE IF ANY BUS COLLISIONS
   WOULD OCCUR ON THE BUS *)

    VAR
        I, J, HI: INTEGER;
        SUM1, SUM2 : REAL;
        PD : REAL;

    BEGIN

        SUM1 := 0;
        SUM2 := 0;
        ACOLL := 0;
        PD := 0.0;
        TEMP := THOST.SLTIME;


        WITH THOST DO
        WITH TERM[TMIT1.TNUM] DO
        BEGIN

        IF (HNUM > NOHOSTS) THEN
        BEGIN
                HI := HNUM - NOHOSTS;
                FOR I:= 1 TO NODIST DO
                IF I <> HI THEN
                IF (DIS[I].NOTX = 0) THEN
                BEGIN
                IF (DIS[I].REFTIME-CURTIME)<PDELAY[HNUM,NOHOSTS+I] THEN
                        BEGIN
                        DIS[I].DIST[1].TERM[1].TXTIME := DIS[I].REFTIME;
                        COLLISION(DIS[I].DIST[1].TERM[1],TERM[TMIT1.TNUM],
                                SUM1,SUM2);

                        IF (ACOLL = 0) THEN
                             ACOLL := 1;
                        IF PD < PDELAY[HNUM,NOHOSTS+I] THEN
                                PD := PDELAY[HNUM,NOHOSTS+I];
                        IF DIS[I].DIST[1].TERM[1].NUMCOLS = 0 THEN
                             DIS[I].REFTIME := CLOCK+9.6E-6+32/EBUSRATE
                        ELSE
                          BEGIN
                             DIS[I].REFTIME := DIS[I].REFTIME + SUM1;
                             COLLIDE(SUM1,DIS[I].DIST[1],1);
                          END;
(*WRITELN(OUT,'COLL TM=',TMIT1.HNUM:2,' W/ DS=',I:2,' RD=',DIS[I].REFDUMP:1,
        ' N/T=',DIS[I].REFTIME);*)
                        END
                ELSE
                IF (DIS[I].DUMPTIME-CURTIME)<PDELAY[HNUM,NOHOSTS+I] THEN
                        BEGIN
                        DIS[I].DIST[2].TERM[1].TXTIME := DIS[I].DUMPTIME;
                        COLLISION(DIS[I].DIST[2].TERM[1],TERM[TMIT1.TNUM],
                                SUM1,SUM2);

                        IF (ACOLL = 0) THEN
                             ACOLL := 1;
                        IF PD < PDELAY[HNUM,NOHOSTS+I] THEN
                                PD := PDELAY[HNUM,NOHOSTS+I];
                        IF DIS[I].DIST[2].TERM[1].NUMCOLS = 0 THEN
                             DIS[I].DUMPTIME := CLOCK+9.6E-6+32/EBUSRATE
                        ELSE
                          BEGIN
                             DIS[I].DUMPTIME := DIS[I].DUMPTIME + SUM1;
                             COLLIDE(SUM1,DIS[I].DIST[2],1);
                          END;
(*WRITELN(OUT,'COLL TM=',TMIT1.HNUM:2,' W/ DS=',I:2,' RD=',DIS[I].REFDUMP:1,
        ' N/T=',DIS[I].DUMPTIME);       *)
                      END;
                END
                ELSE
                IF (DIS[I].DIST[DIS[I].REFDUMP].TERM[1].TXTIME-CURTIME)
                    < PDELAY[HNUM,NOHOSTS+I] THEN
                        BEGIN
                        JUNK := DIS[I].REFDUMP;
                        COLLISION(DIS[I].DIST[JUNK].TERM[1],TERM[TMIT1.TNUM],
                                SUM1,SUM2);

                        IF (ACOLL = 0) THEN
                             ACOLL := 1;
                        IF PD < PDELAY[HNUM,NOHOSTS+I] THEN
                                PD := PDELAY[HNUM,NOHOSTS+I];
                        IF DIS[I].DIST[JUNK].TERM[1].NUMCOLS = 0 THEN
                             DIS[I].DIST[JUNK].TERM[1].TXTIME :=
                                CLOCK + 9.6E-6 + 32/EBUSRATE
                        ELSE
                          BEGIN
                              DIS[I].DIST[JUNK].TERM[1].TXTIME := DIS[I].
                                    DIST[JUNK].TERM[1].TXTIME + SUM1;
                              COLLIDE(SUM1,DIS[I].DIST[JUNK],1);
                          END;
(*WRITELN(OUT,'COLL TM=',TMIT1.HNUM:2,' W/ DS=',I:2,' RD=',DIS[I].REFDUMP:1,
        ' N/T=',DIS[I].DIST[JUNK].TERM[1].TXTIME);     *)
                        END;

                FOR I:= 1 TO NOHOSTS DO
                   FOR J:= 1 TO HOST[I].NOTERMS DO
                        IF (HOST[I].TERM[J].TXTIME-CURTIME) <
                                       PDELAY[HNUM,I] THEN
                        BEGIN
                        COLLISION(HOST[I].TERM[J],TERM[TMIT1.TNUM],
                                SUM1,SUM2);

                        IF (ACOLL = 0) THEN
                             ACOLL := 1;
                        IF PD < PDELAY[HNUM,I] THEN
```

```
                                PD := PDELAY[HNUM,I];
                        IF HOST[I].TERM[J].NUMCOLS = 0 THEN
                             HOST[I].TERM[J].TXTIME :=
                                CLOCK + 9.6E-6 + 32/EBUSRATE
                        ELSE
                          BEGIN
                             HOST[I].TERM[J].TXTIME := HOST[I].TERM[J].TXTIME+
                                SUM1;
                             COLLIDE(SUM1,HOST[I],J);
                          END;
(*WRITELN(OUT,'COLL TM=',TMIT1.HNUM:2,' W/ DS=',I:2,' TT=',J:2,' N/T=',
          HOST[I].TERM[J].TXTIME);      *)
                        END;

        END
    ELSE
        BEGIN
            FOR I:=1 TO NODIST DO
              IF(DIS[I].NOTX = 0) THEN
                BEGIN
                IF (DIS[I].REFTIME-CURTIME)<PDELAY[HNUM,NOHOSTS+I] THEN
                        BEGIN
                        DIS[I].DIST[1].TERM[1].TXTIME := DIS[I].REFTIME;
                        COLLISION(DIS[I].DIST[1].TERM[1],TERM[TMIT1.TNUM],
                                SUM1,SUM2);

                        IF (ACOLL = 0) THEN
                             ACOLL := 1;
                        IF PD < PDELAY[HNUM,NOHOSTS+I] THEN
                                PD := PDELAY[HNUM,NOHOSTS+I];
                        IF DIS[I].DIST[1].TERM[1].NUMCOLS = 0 THEN
                             DIS[I].REFTIME := CLOCK+9.6E-6+32/EBUSRATE
                        ELSE
                          BEGIN
                             DIS[I].REFTIME := DIS[I].REFTIME + SUM1;
                             COLLIDE(SUM1,DIS[I].DIST[1],1);
                          END;
(*WRITELN(OUT,'COLL TM=',TMIT1.HNUM:2,' W/ DS=',I:2,' RD=',DIS[I].REFDUMP:1,
        ' N/T=',DIS[I].REFTIME);        *)
                        END
                ELSE
                IF (DIS[I].DUMPTIME-CURTIME)<PDELAY[HNUM,NOHOSTS+I] THEN
                        BEGIN
                        DIS[I].DIST[2].TERM[1].TXTIME := DIS[I].DUMPTIME;
                        COLLISION(DIS[I].DIST[2].TERM[1],TERM[TMIT1.TNUM],
                                SUM1,SUM2);

                        IF (ACOLL = 0) THEN
                             ACOLL := 1;
                        IF PD < PDELAY[HNUM,NOHOSTS+I] THEN
                                PD := PDELAY[HNUM,NOHOSTS+I];
                        IF DIS[I].DIST[2].TERM[1].NUMCOLS = 0 THEN
                             DIS[I].DUMPTIME := CLOCK+9.6E-6+32/EBUSRATE
                        ELSE
                          BEGIN
                             DIS[I].DUMPTIME := DIS[I].DUMPTIME + SUM1;
                             COLLIDE(SUM1,DIS[I].DIST[2],1);
                          END;
(*WRITELN(OUT,'COLL TM=',TMIT1.HNUM:2,' W/ DS=',I:2,' RD=',DIS[I].REFDUMP:1,
        ' N/T=',DIS[I].DUMPTIME);         *)
                        END;
                END
                ELSE
                IF (DIS[I].DIST[DIS[I].REFDUMP].TERM[1].TXTIME-CURTIME)
                    < PDELAY[HNUM,NOHOSTS+I] THEN
                        BEGIN
                        JUNK := DIS[I].REFDUMP;
                        COLLISION(DIS[I].DIST[JUNK].TERM[1],TERM[TMIT1.TNUM],
                                SUM1,SUM2);

                        IF (ACOLL = 0) THEN
                             ACOLL := 1;
                        IF PD < PDELAY[HNUM,NOHOSTS+I] THEN
                                PD := PDELAY[HNUM,NOHOSTS+I];
                        IF DIS[I].DIST[JUNK].TERM[1].NUMCOLS = 0 THEN
                             DIS[I].DIST[JUNK].TERM[1].TXTIME :=
                                CLOCK + 9.6E-6 + 32/EBUSRATE
                        ELSE
                          BEGIN
                              DIS[I].DIST[JUNK].TERM[1].TXTIME := DIS[I].
                                   DIST[JUNK].TERM[1].TXTIME + SUM1;
                              COLLIDE(SUM1,DIS[I].DIST[JUNK],1);
                          END;
(*WRITELN(OUT,'COLL TM=',TMIT1.HNUM:2,' W/ DS=',I:2,' RD=',DIS[I].REFDUMP:1,
        ' N/T=',DIS[I].DIST[JUNK].TERM[1].TXTIME);      *)
                        END;

                FOR I:= 1 TO NOHOSTS DO
                 IF I <> TMIT1.HNUM THEN
                   FOR J:= 1 TO HOST[I].NOTERMS DO
                        IF (HOST[I].TERM[J].TXTIME-CURTIME) <
                                        PDELAY[HNUM,I] THEN
                        BEGIN
                        COLLISION(HOST[I].TERM[J],TERM[TMIT1.TNUM],
                           SUM1,SUM2);

                        IF (ACOLL = 0) THEN
                             ACOLL := 1;
                        IF PD < PDELAY[HNUM,I] THEN
                                PD := PDELAY[HNUM,I];
                        IF HOST[I].TERM[J].NUMCOLS = 0 THEN
                             HOST[I].TERM[J].TXTIME :=
                                CLOCK + 9.6E-6 + 32/EBUSRATE
                        ELSE
                          BEGIN
                             HOST[I].TERM[J].TXTIME := HOST[I].TERM[J].TXTIME+
                                 SUM1;
                              COLLIDE(SUM1,HOST[I],J);
                           END;
(*WRITELN(OUT,'COLL TM=',TMIT1.HNUM:2,' W/ DS=',I:2,' TT=',J:2,' N/T=',
          HOST[I].TERM[J].TXTIME);      *)
                      END;

        END;

    IF ( ACOLL = 1) THEN
        BEGIN
        TOTCOLS := TOTCOLS + 1;
        BUSBUSY := BUSBUSY + PD + 9.6E-6 + 32/EBUSRATE;
        CLOCK := CLOCK + PD + 9.6E-6 + 32/EBUSRATE;
        IF TERM[TMIT1.TNUM].NUMCOLS = 0 THEN
            TXTIME := CLOCK
        ELSE
          BEGIN
            TXTIME := TXTIME + SUM2;
            COLLIDE(SUM2,THOST,TMIT1.TNUM);
          END;
(*WRITELN(OUT,'N/T TM=',TXTIME,' D=',TMIT1.HNUM:2,' #COLS=',TMIT1.NUMCOLS);*)
        END;

   END;
END;

(******************************************************************************)


 PROCEDURE DEFER (SUM : REAL;VAR CHOST : HOSTRECORD;TNUM : INTEGER);


(******************************************************************************)


BEGIN

(* UPDATES STATION PARAMETERS TO INDICATE DEFER *)

WITH CHOST DO
BEGIN
    WTTIME := WTTIME + SUM;
    NOWAITS := NOWAITS + 1;
    IF ((MINWAIT > SUM) AND (SUM <> 0.0)) THEN MINWAIT := SUM;
    IF (MAXWAIT < SUM) THEN
        BEGIN
                MAXWAIT := SUM;
                MAXRCVRWT := TERM[TNUM].RCVR;
        END;

END;
```

```
END;

(*****************************************************************************)


 PROCEDURE CHECKDEFER(VAR THOST : HOSTRECORD);


(*****************************************************************************)


(* COMPARE TRANSMITTER WITH OTHER STATIONS TO SEE IF THAT STATION MUST
   DEFER PACKET TRANSMISSION UNTIL AFTER THE CURRENT TRANSMISSION IS COMPLETE*)


   VAR
        I, J, HI: INTEGER;
        SUM : REAL;
        CHECK : REAL;


    BEGIN

        WITH THOST DO
        WITH TERM[TMIT1.TNUM] DO


        IF (HNUM > NOHOSTS) THEN
        BEGIN
                HI := HNUM - NOHOSTS;
                FOR I:= 1 TO NODIST DO
                IF I <> HI THEN
                IF (DIS[I].NOTX = 0) THEN
                BEGIN
                CHECK := DIS[I].REFTIME - TXTIME;
                IF (CHECK  < (SLTIME + PDELAY[HNUM,NOHOSTS+I]+9.6E-6)) THEN
                        BEGIN
(* WRITELN(OUT,'DEFER T=',TMIT1.HNUM:2,' D=',I:2,' RD=',DIS[I].REFDUMP:1);*)
                                DIS[I].REFTIME := TXTIME + SLTIME;
                                SUM := 0.0;
                                IF CHECK > 0.0 THEN
                                   SUM := SLTIME - CHECK + 9.6E-6 +
                                        PDELAY[HNUM,NOHOSTS+I];
                                IF SUM < 0.0 THEN SUM := 0.0;
                                DEFER(SUM,DIS[I].DIST[1],1);
                        END
                        ELSE
                        BEGIN
                        CHECK := DIS[I].DUMPTIME - TXTIME;
                        IF (CHECK<(SLTIME+PDELAY[HNUM,NOHOSTS+I]+9.6E-6))THEN
                        BEGIN
```

```
(*WRITELN(OUT,'DEFER T=',TMIT1.HNUM:2,' WITH D=',I:2,' PD=',DIS[I].REFDUMP:1);*)
                                DIS[I].DUMPTIME := TXTIME + SLTIME;
                                SUM := 0.0;
                                IF CHECK >0.0 THEN
                                   SUM := SLTIME - CHECK + 9.6E-6 +
                                        PDELAY[HNUM,NOHOSTS+I];
                                IF SUM < 0.0 THEN SUM := 0.0;
                                DEFER(SUM,DIS[I].DIST[2],1);
                        END;
                      END;
                END
                ELSE
                BEGIN
                CHECK := DIS[I].DIST[DIS[I].REFDUMP].TERM[1].TXTIME-TXTIME;
                IF CHECK < (SLTIME+PDELAY[HNUM,NOHOSTS+I]+9.6E-6) THEN
                        BEGIN
(*WRITELN(OUT,'DEFER T=',TMIT1.HNUM:2,' WITH D=',I:2,' RD=',DIS[I].REFDUMP:1);*)
                        DIS[I].DIST[DIS[I].REFDUMP].TERM[1].TXTIME:=
                        TXTIME + SLTIME;
                        SUM := 0.0;
                        IF CHECK > 0.0 THEN
                           SUM := SLTIME - CHECK+9.6E-6+PDELAY[HNUM,NOHOSTS+I];
                        IF SUM < 0.0 THEN SUM := 0.0;
                        DEFER(SUM,DIS[I].DIST[DIS[I].REFDUMP],1);
                        END;
                  END;

                FOR I:= 1 TO NOHOSTS DO
                   FOR J:= 1 TO HOST[I].NOTERMS DO
                        BEGIN
                        CHECK := HOST[I].TERM[J].TXTIME - TXTIME;
                        IF (CHECK < (SLTIME + PDELAY[HNUM,I]+9.6E-6)) THEN
                                 BEGIN
(*    WRITELN(OUT,'DEFER T=',TMIT1.HNUM:2,' WITH H=',I:2,
                ' TERM= ',J:2,' CK=',CHECK);     *)
                                   HOST[I].TERM[J].TXTIME:=
                                   TXTIME + SLTIME;
                                   SUM := 0.0;
                                   IF CHECK > 0.0 THEN
                                      SUM:=SLTIME-CHECK+9.6E-6+PDELAY[HNUM,I];
                                   IF SUM < 0.0 THEN SUM := 0.0;
                                   DEFER(SUM,HOST[I],J);
                                END;
                        END;

        END
    ELSE
        BEGIN
            FOR I:=1 TO NODIST DO
              IF(DIS[I].NOTX = 0) THEN
                BEGIN
                CHECK := DIS[I].REFTIME - TXTIME;
                IF (CHECK < (SLTIME + PDELAY[HNUM,NOHOSTS+I]+9.6E-6)) THEN
```

```
                         BEGIN
(*WRITELN(OUT,'DEFER T=',TMIT1.HNUM:2,' WITH D=',I:2,' RD=',DIS[I].REFDUMP:1);*)
                                 DIS[I].REFTIME := TXTIME + SLTIME;
                                 SUM := 0.0;
                                 IF CHECK > 0.0 THEN
                                    SUM := SLTIME - CHECK + 9.6E-6 +
                                         PDELAY[HNUM,NOHOSTS+I];
                                 IF SUM < 0.0 THEN SUM := 0.0;
                                 DEFER(SUM,DIS[I].DIST[1],1);
                         END
                         ELSE
                         BEGIN
                         CHECK := DIS[I].DUMPTIME - TXTIME;
                         IF (CHECK < (SLTIME +
                             PDELAY[HNUM,NOHOSTS+I] + 9.6E-6)) THEN
                         BEGIN
(*WRITELN(OUT,'DEFER T=',TMIT1.HNUM:2,' WITH D=',I:2,' RD=',DIS[I].REFDUMP:1);*)
                                 DIS[I].DUMPTIME := TXTIME + SLTIME;
                                 SUM := 0.0;
                                 IF CHECK > 0.0 THEN
                                    SUM := SLTIME - CHECK + 9.6E-6 +
                                         PDELAY[HNUM,NOHOSTS+I];
                                 IF SUM < 0.0 THEN SUM := 0.0;
                                 DEFER(SUM,DIS[I].DIST[2],1);
                         END;
                       END;
                END
                ELSE
                BEGIN
                CHECK := DIS[I].DIST[DIS[I].REFDUMP].TERM[1].TXTIME-TXTIME;
                IF (CHECK < (SLTIME+PDELAY[HNUM,NOHOSTS+I]+9.6E-6)) THEN
                         BEGIN
(*WRITELN(OUT,'DEFER T=',TMIT1.HNUM:2,' WITH D=',I:2,' RD=',DIS[I].REFDUMP:1);*)
                         DIS[I].DIST[DIS[I].REFDUMP].TERM[1].TXTIME :=
                         TXTIME + SLTIME;
                         SUM := 0.0;
                         IF CHECK > 0.0 THEN
                            SUM := SLTIME - CHECK + 9.6E-6 +
                                 PDELAY[HNUM,NOHOSTS+I];
                         IF SUM < 0.0 THEN SUM := 0.0;
                         DEFER(SUM,DIS[I].DIST[DIS[I].REFDUMP],1);
                         END;
                     END;

                FOR I:= 1 TO NOHOSTS DO
                   IF(I<>TMIT1.HNUM) THEN
                   FOR J:= 1 TO HOST[I].NOTERMS DO
                         BEGIN
                         CHECK := HOST[I].TERM[J].TXTIME-TXTIME;
                         IF (CHECK< (SLTIME + PDELAY[HNUM,I]+9.6E-6)) THEN
                                BEGIN
(*    WRITELN(OUT,'DEFER T=',TMIT1.HNUM:2,' WITH H=',I:2,
                ' TERM= ',J:2,' CK=',CHECK);    *)
```

```
                                   HOST[I].TERM[J].TXTIME:=
                                   TXTIME • SLTIME;
                                   SUM := 0.0;
                                   IF CHECK > 0.0 THEN
                                      SUM := SLTIME - CHECK + 9.6E-6 +
                                        PDELAY[HNUM,I];
                                   IF SUM < 0.0 THEN SUM := 0.0;
                                   DEFER(SUM,HOST[I],J);
                                END;
                        END;

        END;
END;
(******************************************************************************)


 PROCEDURE UPDATE;


(******************************************************************************)
(* UPDATE STATION THAT JUST TRANSMITTED A PACKET *)


   VAR
    I,J,K,L : INTEGER;
    SSP: INTEGER;

    BEGIN

        (* COUNT NO PACKETS RECEIVED AT A STATION *)
        IF TMIT1.RCVR > NOHOSTS THEN
         BEGIN
           JUNK := TMIT1.RCVR - NOHOSTS;
           DIS[JUNK].DIST[1].PKTSRX := DIS[JUNK].DIST[1].PKTSRX • 1;
         END
        ELSE
         HOST[TMIT1.RCVR].PKTSRX := HOST[TMIT1.RCVR].PKTSRX + 1;


        (* IF AN OPERATOR REQUEST WAS TRANSMITTED *)
        IF TMIT1.HNUM <= NOHOSTS THEN
                BEGIN
                        WITH HOST[TMIT1.HNUM] DO
                        WITH TERM[TMIT1.TNUM] DO
                          BEGIN

                          PKTSTX := PKTSTX + 1;

                          CLOCK := CURTIME • SLTIME;
                          BUSBUSY := BUSBUSY + SLTIME + 9.6E-6;
```

```
                  USAGE := USAGE + SLTIME;

                FOR I:= 1 TO NODIST-1 DO
                IF TMIT1.RCVR = (NOHOSTS + I) THEN
                BEGIN
                      TXTIME := CLOCK;
                      RCVR := NOHOSTS + I + 1;
                END;

                IF TMIT1.RCVR = NODIST+NOHOSTS THEN
                BEGIN
                      TXTIME := MEANREQ*HCYC[HNUM,TNUM]+
                                (-(MEANREQ/NOTERMS)*LN(RAN(IDUM)));
                      HCYC[HNUM,TNUM] := HCYC[HNUM,TNUM] + 1;
                      RCVR := NOHOSTS+1;
                      NUMCOLS := 0;
                      SP := SP + 1;
                      HTX[SP] := TMIT1.HNUM;
                    FOR I:= 1 TO NODIST DO
                      BEGIN
                      OPREQ[I] := OPREQ[I] + 1;
                      SNOTX[I,SP] := DIS[I].NOTX;
                      SREFDUMP[I,SP] := DIS[I].REFDUMP;
                      FIN[I,SP] := 11;
                      DIS[I].REFDUMP := 1;
                      DIS[I].NOTX := DIS[I].PKTSREF;
                      DIS[I].DIST[1].TERM[SP+1]:=DIS[I].DIST[1].TERM[1];
                      WITH DIS[I].DIST[1].TERM[1] DO
                        BEGIN
                          TXTIME := CLOCK;
                          IF I<>1 THEN TXTIME:=CLOCK;
                          RCVR := TMIT1.HNUM;
                          NUMCOLS := 0;
                        END;
                      END;
                END;
           END;
        END
ELSE
(* A DISTRIBUTOR JUST TRANSMITTED A PACKET - UPDATE FOR NEXT CYCLE *)
        BEGIN
                WITH DIS[TMIT1.HNUM - NOHOSTS] DO
                WITH DIST[DIS[TMIT1.HNUM-NOHOSTS].REFDUMP] DO
                BEGIN

                  NOTX := NOTX -1;
                  PKTSTX := PKTSTX + 1;

                  CLOCK := CURTIME + SLTIME;
                  BUSBUSY := BUSBUSY + SLTIME + 9.6E-6;
                  USAGE := USAGE + SLTIME;
```

```
 (* CHECK TO SEE IF ALL PACKETS HAVE BEEN SENT OUT FOR THE OPERATOR   *)
 (* REQUESTED SCREEN REFRESH - RESTORE DISTRIBUTOR TO NORMAL OPERATING*)
 (* MODE WHEN PACKETS HAVE BEEN TRANSMITTED                           *)

 SSP := 0;
  FOR J:= 1 TO NODIST DO
     IF (DIS[J].NOTX = 0) AND (TMIT1.HNUM = NOHOSTS+J)
          AND (OPREQ[J] <> 0) THEN
      FOR I:=1 TO SP DO
       IF HTX[I] = TMIT1.RCVR THEN
       BEGIN
          DIS[J].REFDUMP := SREFDUMP[J,I];
          DIS[J].DIST[1].TERM[1].NUMCOLS := 0;
          DIS[J].NOTX := SNOTX[J,I];
          DIS[J].DIST[1].TERM[1] := DIS[J].DIST[1].TERM[I+1];
          FIN[J,I] := 1;
          OPREQ[J] := OPREQ[J] - 1;
          SSP := I;
        WITH DIS[J] DO
          IF REFDUMP = 1 THEN
          BEGIN
            IF (NOTX=0) AND (REFTIME<CLOCK) THEN
                  REFTIME := CLOCK
            ELSE
            IF (NOTX<>0) AND (DIST[1].TERM[1].TXTIME<CLOCK) THEN
                  DIST[1].TERM[1].TXTIME := CLOCK;
            IF DUMPTIME < CLOCK THEN
                  DUMPTIME := CLOCK;
          END
          ELSE
          BEGIN
            IF (NOTX=0) AND (DUMPTIME<CLOCK) THEN
                  DUMPTIME := CLOCK
            ELSE
            IF (NOTX<>0) AND (DIST[2].TERM[1].TXTIME<CLOCK) THEN
                  DIST[2].TERM[1].TXTIME := CLOCK;
            IF REFTIME < CLOCK THEN
                  REFTIME := CLOCK;
          END;
       END;

IF SSP <> 0 THEN
   BEGIN
     JUNK := 0;
     FOR L:= 1 TO NODIST DO
          JUNK := JUNK + FIN[L,SSP];
     IF (JUNK=NODIST) THEN
          BEGIN
          FOR L:= 1 TO NODIST DO
                  FIN[L,SSP] := 0;
          J := 0;
          FOR K := 1 TO SP DO
       BEGIN
        JUNK := 0;
        FOR L:= 1 TO NODIST DO
          JUNK := JUNK + FIN[L,K];
        IF (JUNK <> 0) THEN
          BEGIN
               J := J + 1;
               HTX[J] := HTX[K];
               FOR L := 1 TO NODIST DO
               BEGIN
               SREFDUMP[L,J] := SREFDUMP[L,K];
               SNOTX[L,J] := SNOTX[L,K];
               FIN[L,J] := FIN[L,K];
               DIS[L].DIST[1].TERM[J+1] := DIS[L].DIST[1].TERM[K+1];
               END;
           END;
       END;
       SP := SP - 1;
    END;
 END
ELSE

                       (* SET UP TO SERVICE NEXT SCREEN REF OR TAPE  *)
                       (* DUMP WHEN ALL PACKETS HAVE BEEN TRANSMITTED*)
                       (* AND A FORCED SCREEN REF IS NOT OCCURRING   *)
                       IF (NOTX = 0) THEN
                               BEGIN
                               TERM[1].NUMCOLS := 0;
                               IF REFDUMP = 1 THEN
                               BEGIN
                                       REFHST := REFHST + 1;
                                       IF REFHST > NOHOSTS THEN
                                               REFHST := 1;
                                       REFTIME := REFTIME +
                                               MEANREF/NOHOSTS;
                               END
                                    ELSE
                                       BEGIN
                                       DUMPHST := DUMPHST + 1;
                                       IF DUMPHST > NOHOSTS THEN
                                               DUMPHST := 1;
                                       DUMPTIME := DUMPTIME +
                                               MEANDUMP/NOHOSTS;
                                       END;
                               IF REFTIME < CLOCK THEN
                                  REFTIME := CLOCK;
                               IF DUMPTIME < CLOCK THEN
                                  DUMPTIME := CLOCK;
                               END
                       ELSE
                       (* OTHERWISE, TRANSMIT NEXT PACKET AS SOON AS*)
                       (* POSSIBLE OR AS SOON AS HOST BUFFER FILLS  *)
                       WITH TERM[1] DO
```

```
                                        BEGIN
                                        NUMCOLS := 0;
                                        TXTIME := CLOCK;
                                        IF (CURTIME + (BITPKT/IORATE)) >
                                                CLOCK THEN
                                            TXTIME := CURTIME+BITPKT/IORATE;
                                        END;
                        END;
                END;

    END;

(******************************************************************************)


 PROCEDURE PRINTSTATS;

(******************************************************************************)

(* WRITE RUN STATISTICS TO AUXOUT TO BE DISPLAYED ON PRINTER *)


VAR
    I, J : INTEGER;
    T1, T2 : REAL;


    BEGIN
WRITELN(AUXOUT,' ');
WRITELN(AUXOUT,' ');
WRITELN(AUXOUT,' ');
WRITELN(AUXOUT,'SOURCE  WAIT TIME          WAIT TIME         DEFER COLL  ',
        'PKTS  PKTS  MINIMUM PKT      MAXIMUM PKT      MAX WAIT');
WRITELN(AUXOUT,'        DEFER              COLLISION         COUNT COUNT ',
        'TX    RX    WAIT TIME        WAIT TIME        RECEIVER ');


      FOR I := 1 TO NODIST DO
        FOR J := 1 TO 2 DO
            WITH DIS[I].DIST[J] DO
             BEGIN
             IF (MINWAIT = 999.9) THEN MINWAIT := 0.0;
             WRITELN (AUXOUT,I:1,',',J:1,' ':4,WTTIME,' ',COLTIME,' ',
                NOWAITS:4,' ':2,NOCOLS:4,' ':2,PKTSTX:4,' ',PKTSRX:4,'    ',
                MINWAIT,' ',MAXWAIT,' ',MAXRCVRWT);
             TOTPKTSTX := TOTPKTSTX + PKTSTX;
             END;

      FOR I := 1 TO NOHOSTS DO
```

```
        WITH HOST[I] DO
           BEGIN
             IF(MINWAIT = 999.9) THEN MINWAIT := 0.0;
             WRITELN (AUXOUT,I:2,' ':5,WTTIME,' ',COLTIME,' ',
                NOWAITS:4,' ':2,NOCOLS:4,' ':2,PKTSTX:4,' ',PKTSRX:4,'    ',
                MINWAIT,' ',MAXWAIT,' ',MAXRCVRWT);
             TOTPKTSTX := TOTPKTSTX + PKTSTX;
           END;

WRITELN(AUXOUT,' ');
WRITELN(AUXOUT,' ');
WRITELN(AUXOUT,' ');
WRITELN(AUXOUT,'SIMULATION RUN TIME (SECONDS): ',SIMTIME);
WRITELN(AUXOUT,'TOTAL BUS COLLISIONS: ',TOTCOLS);
WRITELN(AUXOUT,'TOTAL PACKETS TX: ',TOTPKTSTX);
WRITELN(AUXOUT,' ');
WRITELN(AUXOUT,' ');
WRITELN(AUXOUT,' AVG BUSBUSY              AVG USAGE             AVG IDLE');
AVGBB := BUSBUSY / TOTPKTSTX;
AVGUS := USAGE / TOTPKTSTX;
IDLE := SIMTIME - BUSBUSY;
AVGID := IDLE / TOTPKTSTX;
WRITELN(AUXOUT,AVGBB,' ':5,AVGUS,' ':5,AVGID);
WRITELN(AUXOUT,' ');
WRITELN(AUXOUT,' TOT BUSBUSY              TOT USAGE             TOT IDLE');
WRITELN(AUXOUT,BUSBUSY,' ':5,USAGE,' ':5,IDLE);
WRITELN(AUXOUT,' ');
SIMTHRUPUT := AVGUS / (AVGBB + AVGID);
WRITELN(AUXOUT,' ');
WRITELN(AUXOUT,' ');
WRITELN(AUXOUT,'S = SIMULATED THROUGHPUT: ',SIMTHRUPUT);
WRITELN(AUXOUT,' ');
OFFLOAD := TOTBITS / SIMTIME;
OFFLOAD := OFFLOAD / EBUSRATE;
WRITELN(AUXOUT,'G = AGGREGATE OFFERED LOAD AS A PERCENT OF BUS CAPACITY: ',
        OFFLOAD);
WRITELN(AUXOUT,' ');
EFFICIENCY := SIMTHRUPUT / OFFLOAD;
WRITELN(AUXOUT,'E = EFFICIENCY (S/G): ',EFFICIENCY);
WRITELN(AUXOUT,' ');
T1 := OFFLOAD * EXP(-(1.27E-9 * OFFLOAD));
T2 := OFFLOAD * (1 + 2 * 1.27E-9) + EXP(-(1.27E-9*OFFLOAD));
TP := T1/ T2;
WRITELN(AUXOUT,'T = THEORETICAL THROUGHPUT: ',TP);

    END;

(*******************************************************************************)

PROCEDURE CHARTRESULTS;
```

```
(*****************************************************************************)

(* WRITE CONDENSED FORM OF RUN STATISTICS TO SF1 AND SF2 TO BE COMPARED WITH
   PREVIOUS RUN RESULTS *)

VAR
        KT : INTEGER;
        IR, BP1, BP2, HI, HB : REAL;
        I, J, PR, PD, HN : INTEGER;
        INDAT: CHAR;
        TDW, TCW, MINW, MAXW : REAL;

BEGIN

RESET(SF1);
IF EOF(SF1) THEN
        BEGIN
        REWRITE(SF1);
        WRITELN(SF1,'                                                            ',
                    'SIMULATION RUN PARAMETERS ');
        WRITELN(SF1,' ');
        WRITELN(SF1,'S.T.   - SIMULATION RUN TIME');
        WRITELN(SF1,'E.B.   - ETHERNET BUS ');
        WRITELN(SF1,'M.F.   - MAIN FRAME ');
        WRITELN(SF1,'T.F.D. - TAPE FILE DUMP');
        WRITELN(SF1,'H.     - HOSTS');
        WRITELN(SF1,'T/H    - TERMINALS PER HOST');
        WRITELN(SF1,'S.R.   - SCREEN REFRESH');
        WRITELN(SF1,'O.R.   - OPERATOR REQUEST');
        WRITELN(SF1,' ');
        WRITELN(SF1,' ');
        WRITELN(SF1,' ');

        WRITELN(SF1,'                      I/O RATES                 NUMBER OF   ',
              ' NUMBER PKTS FOR  BITS/PKT FOR                AVG TIMES (SEC)');
        WRITELN(SF1,'RUN  S.T.     E.B.     M.F.     H.       M.F. H. T/H ',
              ' S.R. T.F.D. O.R. S.R.     T.F.D.   O.R.     S.R. T.F.D. O.R.');
        END;

     DAT[1] := 'TEMP                ';
     BIND (DFILE,DAT[1],ISTAT);
     RESET(SF1);
     KT := -14;
     REWRITE (DFILE);
     WHILE NOT EOF(SF1) DO
        BEGIN
            KT := KT + 1;
            WHILE NOT EOLN(SF1) DO
                BEGIN
                   READ (SF1,INDAT);
                   WRITE (DFILE,INDAT);
                END;
            READLN(SF1);
            WRITELN(DFILE);
        END;

IR := DIS[1].DIST[1].IORATE;
PR := 0;
PD := 0;
FOR I := 1 TO NODIST DO
 BEGIN
   PR := PR + DIS[I].PKTSREF;
   PD := PD + DIS[I].PKTSDUMP;
  END;
BP1 := DIS[1].DIST[1].BITPKT;
BP2 := DIS[1].DIST[2].BITPKT;
HI := HOST[1].IORATE;
HN := HOST[1].NOTERMS;
HB := HOST[1].BITPKT;

WRITELN(DFILE,KT:3,' ',SIMTIME:5,' ',EBUSRATE:5,' ',IR:5,' ',HI:5,
     ' ',NODIST:2,'   ',NOHOSTS:2,'  ',HN:2,'  ',PR:2,'     ',PD:2,'     1',
     '  ',BP2:5,' ',BP2:5,' ',HB:5,'  ',MEANREF:2,'     ',MEANDUMP:2,
     '    ',MEANREQ:3);
CLOSE(DFILE);
BIND(DFILE,DAT[1],ISTAT);
     RESET(DFILE);
     REWRITE (SF1);
     WHILE NOT EOF(DFILE) DO
        BEGIN
            WHILE NOT EOLN(DFILE) DO
                BEGIN
                   READ (DFILE,INDAT);
                   WRITE (SF1,INDAT);
                END;
            READLN(DFILE);
            WRITELN(SF1);
        END;

     CLOSE (DFILE);

RESET(SF2);
IF EOF(SF2) THEN
        BEGIN
        REWRITE(SF2);
        WRITELN(SF2,'                                                     ',
                'SIMULATION RUN RESULTS ');
        WRITELN(SF2,' ');
        WRITELN(SF2,'G   - AGGREGRATE OFFERED LOAD AS A PERCENT OF BUS',
              ' CAPACITY');
        WRITELN(SF2,'S   - SIMULATED THROUGHPUT');
        WRITELN(SF2,'T   - THEORETICAL THROUGHPUT ');
        WRITELN(SF2,'E   - EFFICIENCY');
        WRITELN(SF2,'TOT WAIT DEFER TIME  - MAXIMUM TOTAL WAITING TIME OF A ',
                    'DEVICE DUE TO PACKETS BEING DEFFERED');
        WRITELN(SF2,'TOT WAIT COLL TIME    - MAXIMUM TOTAL WAITING TIME OF A ',
                   'DEVICE DUE TO PACKET COLLISIONS ');
        WRITELN(SF2,'MIN WAIT PACKET       - MINIMUM PACKET WAIT TIME TO ACCESS',
                    ' ETHERNET BUS');
        WRITELN(SF2,'MAX WAIT PACKET       - MAXIMUM PACKET WAIT TIME TO ACCESS',
                    ' ETHERNET BUS ');
        WRITELN(SF2,' ');
        WRITELN(SF2,' ');
        WRITELN(SF2,' ');


        WRITELN(SF2,'                                          NUM    PKTS',
              '    TOT WAIT      TOT WAIT       MIN WAIT   MAX WAIT  ');
        WRITELN(SF2,'RUN  G        S        T        E          COLS   TMIT',
              '    DEFER TIME    COLL TIME      PACKET     PACKET   ');
        END;

     DAT[1] := 'TEMP              ';
     BIND (DFILE,DAT[1],ISTAT);
     RESET(SF2);
     KT := -14;
     REWRITE (DFILE);
     WHILE NOT EOF(SF2) DO
        BEGIN
            KT := KT + 1;
            WHILE NOT EOLN(SF2) DO
                BEGIN
                   READ (SF2,INDAT);
                   WRITE (DFILE,INDAT);
                END;
            READLN(SF2);
            WRITELN(DFILE);
        END;

TDW := 0.0;
TCW := 0.0;
MINW := 999.9;
MAXW := 0.0;
FOR I:= 1 TO NODIST DO
 FOR J:= 1 TO 2 DO
   BEGIN
     IF TDW < DIS[I].DIST[J].WTTIME THEN TDW := DIS[I].DIST[J].WTTIME;
     IF TCW < DIS[I].DIST[J].COLTIME THEN TCW := DIS[I].DIST[J].COLTIME;
     IF ((MINW>DIS[I].DIST[J].MINWAIT)AND(DIS[I].DIST[J].MINWAIT<>0.0)) THEN
          MINW := DIS[I].DIST[J].MINWAIT;
     IF MINW = 999.9 THEN MINW := 0.0;
     IF MAXW < DIS[I].DIST[J].MAXWAIT THEN MAXW := DIS[I].DIST[J].MAXWAIT;
   END;


 FOR I:= 1 TO NOHOSTS DO
   BEGIN
```

```
     IF TDW < HOST[I].WTTIME THEN TDW := HOST[I].WTTIME;
     IF TCW < HOST[I].COLTIME THEN TCW := HOST[I].COLTIME;
     IF ((MINW>HOST[I].MINWAIT)AND(HOST[I].MINWAIT<>0.0)) THEN
          MINW := HOST[I].MINWAIT;
     IF MAXW < HOST[I].MAXWAIT THEN MAXW := HOST[I].MAXWAIT;
   END;


WRITELN(DFILE,KT:3,' ',OFFLOAD:5,' ',SIMTHRUPUT:5,' ',TP:5,' ',EFFICIENCY:5,
        '     ',TOTCOLS:4,'    ',TOTPKTSTX:5,'    ',TDW:5,'        ',TCW:5,'         ',
        MINW:5,'   ',MAXW:5);

CLOSE(DFILE);
BIND(DFILE,DAT[1],ISTAT);
     RESET(DFILE);
     REWRITE (SF2);
     WHILE NOT EOF(DFILE) DO
        BEGIN
            WHILE NOT EOLN(DFILE) DO
                BEGIN
                   READ (DFILE,INDAT);
                   WRITE (SF2,INDAT);
                END;
            READLN(DFILE);
            WRITELN(SF2);
        END;

     CLOSE (DFILE);

END;

(*****************************************************************************)

(* MAIN PROCEDURE                                                            *)


  BEGIN

(*****************************************************************************)


     INITIALIZE;

     CONFIGURE;

(* IF OPERATOR SELECTED OPTION #4 IN CONFIGURE THEN RUN SIMULATION
   OTHERWISE ABORT SIMULATION IMMEDIATELY                    *)
IF SL = 4 THEN
BEGIN
```

```
     PRINTCONFIG;

     CLOCK := 9.6E-6;

     REPEAT

        FINDNEXT(TMIT1);

        CURTIME := TMIT1.TXTIME;

        IF (CURTIME < (CLOCK + 9.6E-6)) THEN
                CURTIME := CLOCK + 9.6E-6;


        IF TMIT1.HNUM > NOHOSTS THEN
           BEGIN
              JUNK := TMIT1.HNUM - NOHOSTS;
              TOTBITS:=TOTBITS+DIS[JUNK].DIST[DIS[JUNK].REFDUMP].BITPKT;
              ACOLLISION(DIS[JUNK].DIST[DIS[JUNK].REFDUMP]);
       IF ACOLL = 0 THEN
          WRITELN(OUT,'TMR=',TMIT1.HNUM:2,' RD=',DIS[JUNK].REFDUMP:1,
              ' RC=',TMIT1.RCVR:2,' TXTM=',TMIT1.TXTIME,' CL=',CLOCK);
            END
         ELSE
            BEGIN
              TOTBITS := TOTBITS + HOST[TMIT1.HNUM].BITPKT;
              ACOLLISION(HOST[TMIT1.HNUM]);
     IF ACOLL = 0 THEN
        WRITELN(OUT,'TMR=',TMIT1.HNUM:2,
           ' RC=',TMIT1.RCVR:2,' TXTM=',TMIT1.TXTIME,' CL=',CLOCK);
            END;

         IF (ACOLL = 0) THEN
              BEGIN
                   IF TMIT1.HNUM > NOHOSTS THEN
                        BEGIN
                          JUNK := TMIT1.HNUM - NOHOSTS;
                          CHECKDEFER(DIS[JUNK].DIST[DIS[JUNK].REFDUMP]);
                        END
                        ELSE
                          CHECKDEFER(HOST[TMIT1.HNUM]);

                   UPDATE;
              END;

(*      WRITELN(OUT,' ');        *)
        UNTIL ( CLOCK >= SIMTIME);

        PRINTSTATS;

        WRITELN ('END OF RUN');

        REPEAT
```

```
            WRITELN(' ');
            WRITELN('ADD RUN INFORMATION TO SUMMARY CHARTS? ENTER "Y" OR "N":');
            READLN(OPCHAR);
         UNTIL ((OPCHAR = 'Y') OR (OPCHAR = 'N'));

         IF OPCHAR = 'Y' THEN
            CHARTRESULTS;

END;
END.
```

APPENDIX II.

PARAMETERS AND RESULTS OF VARIOUS SCENARIOS

Runs 1 - 10	BASIC CONFIGURATION	
	Simulation Run Time:	5 sec
	Ethernet Bus Rate:	10 Mbits/sec
	Number of Main Frames:	2
	Main Frame I/O Rate:	10 Mbits/sec
	Number of Hosts:	20
	Host I/O Rate:	2.4 Mbits/sec
	Number of Terminals per Host:	10
	Screen Refresh	7680 bits/packet
	Tape File Dump	7680 bits/packet
	Operator Request	1024 bits/packet
	Operator Request No. packets:	1 per main frame
	Average Time of Screen Refreshes:	2 sec
Runs 1 - 10	VARIABLE PARAMETERS	
	Number of Packets for Screen Refresh:	2 - 10
	Number of Packets for Tape File Dump:	2 - 10
	Average Time of Tape File Dumps:	4, 10, or 15 sec
	Average Time for Operator Requests:	10, 18, 180 sec

CONTENTS

```
            ETHERNET SIMULATION PARAMETERS  RUN 1


ETHERNET BUS IO RATE:  1.000000000E+07

NUMBER OF MAIN FRAMES:          2
NUMBER OF HOSTS:         20

REFRESHES TO HOSTS IN SECONDS:  2.000000000E+00
DUMPS TO HOSTS IN SECONDS:  1.500000000E+01
OPERATOR REFRESH REQUESTS IN SECONDS:  1.800000000E+02


MAIN FRAME NUMBER:          1

I/O RATE OF MAIN FRAME:  1.000000000E+07
DISTANCE FROM REFRENCE POINT:  4.000000000E+02
FIRST REFRESH TIME:  1.000000000E-01 TO HOST:  1
FIRST DUMP TIME:  7.500000000E-01 TO HOST:  1

REFRESH OPERATION:
     BITS/PACKET:  7.680000000E+03
     SLTIME:  7.680000000E-04
     NUMBER OF PACKETS SENT TO EACH TERMINAL:          1

DUMP OPERATION:
     BITS/PACKET:  7.680000000E+03
     SLTIME:  7.680000000E-04
     NUMBER OF PACKETS SENT TO EACH TERMINAL:          1

MAIN FRAME NUMBER:          2

I/O RATE OF MAIN FRAME:  1.000000000E+07
DISTANCE FROM REFRENCE POINT:  6.750000000E+02
FIRST REFRESH TIME:  1.000000000E-01 TO HOST:  1
FIRST DUMP TIME:  7.500000000E-01 TO HOST:  1

REFRESH OPERATION:
     BITS/PACKET:  7.680000000E+03
     SLTIME:  7.680000000E-04
     NUMBER OF PACKETS SENT TO EACH TERMINAL:          1

DUMP OPERATION:
     BITS/PACKET:  7.680000000E+03
     SLTIME:  7.680000000E-04
     NUMBER OF PACKETS SENT TO EACH TERMINAL:          1


HOST NUMBER:          1
```

```
NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  5.000000000E+01
SLOT TIME:  1.024000000E-04


HOST NUMBER:          2

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  1.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:          3

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  1.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:          4

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  2.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:          5

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  2.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:          6

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  3.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:          7
```

```
NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  3.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:         8

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  4.250000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:         9

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  4.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        10

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  5.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        11

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  5.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        12

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  6.000000000E+02
SLOT TIME:  1.024000000E-04
```

```
HOST NUMBER:        13

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  6.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        14

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  7.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        15

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  7.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        16

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  8.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        17

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  8.250000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        18

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  8.750000000E+02
SLOT TIME:  1.024000000E-04
```

```
HOST NUMBER:       19

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  9.250000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:       20

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  9.750000000E+02
SLOT TIME:  1.024000000E-04
```

```
SOURCE  WAIT TIME         WAIT TIME         DEFER COLL  PKTS  PKTS  MINIMUM PKT       MAXIMUM PKT       MAX WAIT
        DEFER             COLLISION         COUNT COUNT TX    RX    WAIT TIME         WAIT TIME         RECEIVER
 1,1    5.071740195E-02   5.776636380E-02    86    166   520   29   1.281828508E-05   5.869193929E-03       2
 1,2    3.605608740E-03   7.491983035E-03     9      9    60    0   3.439916077E-05   2.918560678E-03       1
 2,1    3.902580916E-02   9.033029368E-03    68    138   519   29   9.949237381E-06   7.756503477E-04      10
 2,2    8.514957607E-03   8.126449837E-03    18     37    60    0   9.949077310E-06   2.043941582E-03       4
  1     0.000000000E+00   0.000000000E+00     0      0     6   86   0.000000000E+00   0.000000000E+00       0
  2     0.000000000E+00   0.000000000E+00     0      0     4   84   0.000000000E+00   0.000000000E+00       0
  3     4.203487519E-03   2.746038678E-03     7      7     2   52   4.086294407E-05   2.046792090E-03      22
  4     0.000000000E+00   0.000000000E+00     0      0     2   82   0.000000000E+00   0.000000000E+00       0
  5     0.000000000E+00   0.000000000E+00     0      0     2   82   0.000000000E+00   0.000000000E+00       0
  6     0.000000000E+00   0.000000000E+00     0      0     4   54   0.000000000E+00   0.000000000E+00       0
  7     7.009777338E-04   0.000000000E+00     1      0     2   62   7.009777338E-04   7.009777338E-04      21
  8     0.000000000E+00   0.000000000E+00     0      0     2   62   0.000000000E+00   0.000000000E+00       0
  9     0.000000000E+00   0.000000000E+00     0      0     2   62   0.000000000E+00   0.000000000E+00       0
 10     3.735443596E-03   1.121998570E-02     8      9     6   47   3.278764756E-05   4.977577060E-03      22
 11     0.000000000E+00   0.000000000E+00     0      0     2   42   0.000000000E+00   0.000000000E+00       0
 12     0.000000000E+00   0.000000000E+00     0      0     2   42   0.000000000E+00   0.000000000E+00       0
 13     8.000583261E-05   0.000000000E+00     1      0     6   46   8.000583261E-05   8.000583261E-05      21
 14     0.000000000E+00   0.000000000E+00     0      0     2   42   0.000000000E+00   0.000000000E+00       0
 15     1.226567748E-03   1.297117913E-04     2      3     6   46   1.287314034E-05   7.254111539E-04      22
 16     3.035164311E-04   4.874695280E-05     1      2     2   42   2.251622156E-05   3.035164311E-04      21
 17     0.000000000E+00   0.000000000E+00     0      0     2   42   0.000000000E+00   0.000000000E+00       0
 18     0.000000000E+00   0.000000000E+00     0      0     0   40   0.000000000E+00   0.000000000E+00       0
 19     1.340129429E-03   3.268968414E-05     2      1     2   42   3.268968414E-05   7.773869808E-04      21
 20     0.000000000E+00   0.000000000E+00     0      0     2   42   0.000000000E+00   0.000000000E+00       0


SIMULATION RUN TIME (SECONDS):  5.000000000E+00
TOTAL BUS COLLISIONS:        182
TOTAL PACKETS TX:       1217

 AVG BUSBUSY           AVG USAGE            AVG IDLE
 7.478450034E-04     7.362787172E-04     3.360618431E-03

 TOT BUSBUSY           TOT USAGE            TOT IDLE
 9.101273691E-01     8.960511989E-01     4.089872631E+00


S = SIMULATED THROUGHPUT:  1.792102398E-01

G = AGGREGATE OFFERED LOAD AS A PERCENT OF BUS CAPACITY:  2.063667192E-01

E = EFFICIENCY (S/G):  8.684066910E-01

T = THEORETICAL THROUGHPUT:  1.710646654E-01
```

```
                     ETHERNET SIMULATION PARAMETERS  RUN 2


ETHERNET BUS IO RATE:  1.000000000E+07

NUMBER OF MAIN FRAMES:          2
NUMBER OF HOSTS:       20

REFRESHES TO HOSTS IN SECONDS:  2.000000000E+00
DUMPS TO HOSTS IN SECONDS:  1.000000000E+01
OPERATOR REFRESH REQUESTS IN SECONDS:  1.000000000E+01


MAIN FRAME NUMBER:          1

I/O RATE OF MAIN FRAME:  1.000000000E+07
DISTANCE FROM REFRENCE POINT:  4.000000000E+02
FIRST REFRESH TIME:  1.000000000E-01 TO HOST:  1
FIRST DUMP TIME:  5.000000000E-01 TO HOST:  1

REFRESH OPERATION:
     BITS/PACKET:  7.680000000E+03
     SLTIME:  7.680000000E-04
     NUMBER OF PACKETS SENT TO EACH TERMINAL:          1

DUMP OPERATION:
     BITS/PACKET:  7.680000000E+03
     SLTIME:  7.680000000E-04
     NUMBER OF PACKETS SENT TO EACH TERMINAL:          1

MAIN FRAME NUMBER:          2

I/O RATE OF MAIN FRAME:  1.000000000E+07
DISTANCE FROM REFRENCE POINT:  6.750000000E+02
FIRST REFRESH TIME:  1.000000000E-01 TO HOST:  1
FIRST DUMP TIME:  5.000000000E-01 TO HOST:  1

REFRESH OPERATION:
     BITS/PACKET:  7.680000000E+03
     SLTIME:  7.680000000E-04
     NUMBER OF PACKETS SENT TO EACH TERMINAL:          1

DUMP OPERATION:
     BITS/PACKET:  7.680000000E+03
     SLTIME:  7.680000000E-04
     NUMBER OF PACKETS SENT TO EACH TERMINAL:          1


HOST NUMBER:          1
```

```
NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  5.000000000E+01
SLOT TIME:  1.024000000E-04


HOST NUMBER:        2

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  1.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        3

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  1.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        4

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  2.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        5

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  2.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        6

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  3.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        7
```

```
NUMBER OF TERMINALS:       10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  3.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        8

NUMBER OF TERMINALS:       10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  4.250000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        9

NUMBER OF TERMINALS:       10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  4.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:       10

NUMBER OF TERMINALS:       10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  5.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:       11

NUMBER OF TERMINALS:       10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  5.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:       12

NUMBER OF TERMINALS:       10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  6.000000000E+02
SLOT TIME:  1.024000000E-04
```

```
HOST NUMBER:        13

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  6.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        14

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  7.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        15

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  7.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        16

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  8.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        17

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  8.250000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        18

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  8.750000000E+02
SLOT TIME:  1.024000000E-04
```

```
HOST NUMBER:       19

NUMBER OF TERMINALS:       10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  9.250000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:       20

NUMBER OF TERMINALS:       10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  9.750000000E+02
SLOT TIME:  1.024000000E-04
```

```
SOURCE  WAIT TIME         WAIT TIME         DEFER COLL  PKTS  PKTS  MINIMUM PKT       MAXIMUM PKT       MAX WAIT
        DEFER             COLLISION         COUNT COUNT TX    RX    WAIT TIME         WAIT TIME         RECEIVER
1,1     8.332189795E-02   3.361753720E-02   137   287   649   159   1.707173656E-06   7.535614227E-03          3
1,2     1.803956111E-02   1.595804851E-02    33    81    91     0   1.292247421E-05   2.824885358E-03          9
2,1     6.732030489E-02   3.771601679E-02   115   269   649   159   1.286123711E-05   6.601658023E-03          1
2,2     2.669748965E-02   7.147383328E-02    53    81    90     0   9.949251933E-06   2.627753057E-02          4
  1     3.155810142E-03   6.327354393E-04     5     6    12    92   2.776261401E-05   7.341254851E-04         22
  2     5.632568485E-04   2.024623949E-04     2     4    16    96   1.751138511E-05   4.851893302E-04         21
  3     5.784542713E-03   8.960172008E-04    13    11    16    96   3.131282361E-05   7.593228630E-04         21
  4     4.905620460E-03   1.859131942E-03    13    14    18    98   5.294998690E-06   8.997215034E-04         21
  5     6.320295539E-03   8.366737484E-04    11    12    20   100   1.729569282E-05   7.650161521E-04         21
  6     6.348252387E-03   9.054515025E-03    12    13    18    98   1.434254408E-05   3.756869426E-03         22
  7     9.108139222E-03   9.999204583E-03    18    21    18    98   2.667092182E-05   2.505174310E-03         21
  8     9.782967431E-05   2.005129793E-04     1     3    12    92   1.679131131E-05   1.149136649E-04         22
  9     5.292978958E-04   2.727476600E-04     2     5    14    94   1.896077651E-05   5.103371193E-04         21
 10     1.928733758E-03   5.366453442E-04     5     5    14    55   5.309025646E-05   7.700742678E-04         21
 11     2.533163647E-03   7.851172390E-04     7    10    14    54   1.304494843E-05   7.702577593E-04         22
 12     6.060944751E-03   3.517639331E-03    13    12    16    56   4.153342668E-05   9.488173565E-04         22
 13     6.451369538E-03   1.818487094E-02    14    14    16    56   2.835091043E-05   9.943897296E-03         21
 14     2.384308115E-03   2.356980114E-03     7     7    20    60   2.777471099E-05   1.409533800E-03         22
 15     4.990169648E-03   9.631570269E-04    10    13    18    58   2.246122289E-05   7.327530241E-04         21
 16     8.975607921E-04   4.944231695E-04     3     7    16    56   2.448918854E-05   7.665412865E-04         21
 17     3.357212260E-03   8.937364143E-03     9    10    12    52   2.991670361E-05   5.925715091E-03         21
 18     2.690919932E-03   5.671680375E-03     6     7    14    54   4.465849741E-05   3.220680723E-03         22
 19     6.554432041E-03   1.041026816E-02    15    17    16    56   2.205641180E-05   2.572287192E-03         21
 20     0.000000000E+00   1.094860273E-04     0     3    18    58   1.921326040E-05   5.521616186E-05         22


SIMULATION RUN TIME (SECONDS):  5.000000000E+00
TOTAL BUS COLLISIONS:        431
TOTAL PACKETS TX:       1797

 AVG BUSBUSY           AVG USAGE           AVG IDLE
 6.629678938E-04     6.502143561E-04     2.119447243E-03

 TOT BUSBUSY           TOT USAGE           TOT IDLE
 1.191353305E+00     1.168435198E+00     3.808646695E+00


S = SIMULATED THROUGHPUT:  2.336870396E-01

G = AGGREGATE OFFERED LOAD AS A PERCENT OF BUS CAPACITY:  2.930995184E-01

E = EFFICIENCY (S/G):  7.972958839E-01

T = THEORETICAL THROUGHPUT:  2.266643163E-01
```

ETHERNET SIMULATION PARAMETERS RUN 3

```
ETHERNET BUS IO RATE:  1.000000000E+07

NUMBER OF MAIN FRAMES:        2
NUMBER OF HOSTS:       20

REFRESHES TO HOSTS IN SECONDS:  2.000000000E+00
DUMPS TO HOSTS IN SECONDS:  4.000000000E+00
OPERATOR REFRESH REQUESTS IN SECONDS:  1.000000000E+01


MAIN FRAME NUMBER:        1

I/O RATE OF MAIN FRAME:  1.000000000E+07
DISTANCE FROM REFRENCE POINT:  4.000000000E+02
FIRST REFRESH TIME:  1.000000000E-01 TO HOST:  1
FIRST DUMP TIME:  2.000000000E-01 TO HOST:  1

REFRESH OPERATION:
     BITS/PACKET:  7.680000000E+03
     SLTIME:  7.680000000E-04
     NUMBER OF PACKETS SENT TO EACH TERMINAL:        1

DUMP OPERATION:
     BITS/PACKET:  7.680000000E+03
     SLTIME:  7.680000000E-04
     NUMBER OF PACKETS SENT TO EACH TERMINAL:        1

MAIN FRAME NUMBER:        2

I/O RATE OF MAIN FRAME:  1.000000000E+07
DISTANCE FROM REFRENCE POINT:  6.750000000E+02
FIRST REFRESH TIME:  1.000000000E-01 TO HOST:  1
FIRST DUMP TIME:  2.000000000E-01 TO HOST:  1

REFRESH OPERATION:
     BITS/PACKET:  7.680000000E+03
     SLTIME:  7.680000000E-04
     NUMBER OF PACKETS SENT TO EACH TERMINAL:        1

DUMP OPERATION:
     BITS/PACKET:  7.680000000E+03
     SLTIME:  7.680000000E-04
     NUMBER OF PACKETS SENT TO EACH TERMINAL:        1


HOST NUMBER:        1
```

```
NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  5.000000000E+01
SLOT TIME:  1.024000000E-04


HOST NUMBER:         2

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  1.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:         3

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  1.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:         4

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  2.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:         5

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  2.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:         6

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  3.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:         7
```

```
NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  3.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        8

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  4.250000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        9

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  4.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:       10

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  5.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:       11

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  5.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:       12

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  6.000000000E+02
SLOT TIME:  1.024000000E-04
```

```
HOST NUMBER:        13

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  6.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        14

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  7.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        15

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  7.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        16

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  8.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        17

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  8.250000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        18

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  8.750000000E+02
SLOT TIME:  1.024000000E-04
```

```
HOST NUMBER:       19

NUMBER OF TERMINALS:       10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  9.250000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:       20

NUMBER OF TERMINALS:       10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  9.750000000E+02
SLOT TIME:  1.024000000E-04
```

```
SOURCE   WAIT TIME         WAIT TIME         DEFER  COLL   PKTS  PKTS   MINIMUM PKT       MAXIMUM PKT        MAX WAIT
         DEFER             COLLISION         COUNT  COUNT  TX    RX     WAIT TIME         WAIT TIME          RECEIVER
 1,1     7.719918548E-02   2.241036320E-02   125    264    645   154    5.543999314E-06   5.351367216E-03        1
 1,2     1.037471187E-03   7.915988301E-04     5     19    240     C    1.109191815E-04   7.449497700E-04        4
 2,1     6.669378257E-02   5.218322972E-02   111    250    644   154    1.272172429E-05   2.526172145E-02       11
 2,2     9.808639920E-04   5.839341023E-04     4      9    240     C    2.546952098E-05   7.218603770E-04        4
  1      1.101220525E-02   1.340989746E-02    19     20     14   114    2.092001172E-05   8.108566385E-03       22
  2      5.008997796E-03   4.792549968E-03    11     12     16   116    1.407350237E-05   2.818805287E-03       22
  3      2.247267744E-03   3.550125530E-05     4      1     16   116    3.550125530E-05   7.556020647E-04       21
  4      2.108084337E-03   4.227659528E-04     4      4     14   114    1.829226579E-05   7.671906443E-04       21
  5      0.000000000E+00   0.000000000E+00     0      0     12    92    0.000000000E+00   0.000000000E+00        0
  6      6.519926700E-03   2.334153632E-03    12     11     14    94    2.255917359E-05   7.726495643E-04       21
  7      3.676292498E-03   2.801193097E-03     6      7     16    96    5.398657323E-05   2.046792090E-03       22
  8      2.850791630E-03   3.697916059E-04     4      9     18    98    2.221301818E-05   7.686269607E-04       22
  9      3.232604941E-03   1.834889601E-04     5      4     16    96    2.478304061E-05   7.283088296E-04       22
 10      3.990453224E-03   3.444067513E-03     8      9     18    79    3.168200399E-05   2.017099096E-03       22
 11      4.970544623E-03   1.530069807E-03     9      9     16    76    2.177188155E-05   7.451581502E-04       22
 12      7.388443633E-03   1.279472221E-02    17     17     16    76    1.504297039E-05   4.594829395E-03       22
 13      2.532082276E-03   4.329680305E-04     5      6     14    74    3.278764211E-05   7.677937385E-04       21
 14      2.371462758E-03   2.575663083E-04     4      5     18    78    1.981573602E-05   7.132479798E-04       22
 15      5.779011647E-03   6.603555678E-03    12     14     20    80    1.505774187E-05   4.080784179E-03       22
 16      3.041221894E-04   2.600452903E-04     2      3     14    74    2.727339743E-05   2.696676067E-04       21
 17      1.804625522E-03   8.153212825E-05     3      2     16    76    2.546951371E-05   7.489086194E-04       22
 18      0.000000000E+00   0.000000000E+00     0      0     12    72    0.000000000E+00   0.000000000E+00        0
 19      1.184257540E-03   2.087466078E-04     4      5     14    74    2.089534478E-05   7.028824595E-04       22
 20      7.781817470E-04   9.042372078E-05     2      2     14    74    2.595708645E-05   6.144717951E-04       21


SIMULATION RUN TIME (SECONDS):  5.000000000E+00
TOTAL BUS COLLISIONS:       331
TOTAL PACKETS TX:      2077


 AVG BUSBUSY             AVG USAGE              AVG IDLE
 6.809916145E-04     6.692976395E-04      1.726326633E-03

 TOT BUSBUSY             TOT USAGE              TOT IDLE
 1.414419593E+00     1.390131197_+00      3.585580417E+00


S = SIMULATED THROUGHPUT:  2.780262394E-01

G = AGGREGATE OFFERED LOAD AS A PERCENT OF BUS CAPACITY:  3.243417581E-01

E = EFFICIENCY (S/G):  8.572014934E-01

T = THEORETICAL THROUGHPUT:  2.449078993E-01
```

```
                ETHERNET SIMULATION PARAMETERS  RUN 4


ETHERNET BUS IO RATE:  1.000000000E+07

NUMBER OF MAIN FRAMES:          2
NUMBER OF HOSTS:          20

REFRESHES TO HOSTS IN SECONDS:  2.000000000E+00
DUMPS TO HOSTS IN SECONDS:  1.500000000E+01
OPERATOR REFRESH REQUESTS IN SECONDS:  1.800000000E+01


MAIN FRAME NUMBER:          1

I/O RATE OF MAIN FRAME:  1.000000000E+07
DISTANCE FROM REFRENCE POINT:  4.000000000E+02
FIRST REFRESH TIME:  1.000000000E-01 TO HOST:  1
FIRST DUMP TIME:  7.500000000E-01 TO HOST:  1

REFRESH OPERATION:
     BITS/PACKET:  7.680000000E+03
     SLTIME:  7.680000000E-04
     NUMBER OF PACKETS SENT TO EACH TERMINAL:          2

DUMP OPERATION:
     BITS/PACKET:  7.680000000E+03
     SLTIME:  7.680000000E-04
     NUMBER OF PACKETS SENT TO EACH TERMINAL:          2

MAIN FRAME NUMBER:          2

I/O RATE OF MAIN FRAME:  1.000000000E+07
DISTANCE FROM REFRENCE POINT:  6.750000000E+02
FIRST REFRESH TIME:  1.000000000E-01 TO HOST:  1
FIRST DUMP TIME:  7.500000000E-01 TO HOST:  1

REFRESH OPERATION:
     BITS/PACKET:  7.680000000E+03
     SLTIME:  7.680000000E-04
     NUMBER OF PACKETS SENT TO EACH TERMINAL:          2

DUMP OPERATION:
     BITS/PACKET:  7.680000000E+03
     SLTIME:  7.680000000E-04
     NUMBER OF PACKETS SENT TO EACH TERMINAL:          2


HOST NUMBER:          1
```

```
NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  5.000000000E+01
SLOT TIME:  1.024000000E-04


HOST NUMBER:         2

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  1.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:         3

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  1.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:         4

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  2.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:         5

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  2.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:         6

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  3.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:         7
```

```
NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  3.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        8

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  4.250000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        9

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  4.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:       10

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  5.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:       11

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  5.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:       12

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  6.000000000E+02
SLOT TIME:  1.024000000E-04
```

```
HOST NUMBER:       13

NUMBER OF TERMINALS:       10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  6.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:       14

NUMBER OF TERMINALS:       10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  7.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:       15

NUMBER OF TERMINALS:       10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  7.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:       16

NUMBER OF TERMINALS:       10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  8.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:       17

NUMBER OF TERMINALS:       10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  8.250000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:       18

NUMBER OF TERMINALS:       10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  8.750000000E+02
SLOT TIME:  1.024000000E-04
```

```
HOST NUMBER:        19

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  9.250000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        20

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  9.750000000E+02
SLOT TIME:  1.024000000E-04
```

```
SOURCE  WAIT TIME         WAIT TIME         DEFER COLL  PKTS  PKTS  MINIMUM PKT       MAXIMUM PKT       MAX WAIT
        DEFER             COLLISION         COUNT COUNT TX     RX   WAIT TIME         WAIT TIME         RECEIVER
1,1     1.786716935E-01   2.423447787E-01   295    568  1225  122   1.286123711E-05   1.960544348E-02        15
1,2     7.025076829E-03   1.433494481E-02    15     33   120    0   1.511410295E-05   5.661376697E-03         1
2,1     1.305904054E-01   1.462587354E-01   227    579  1224  122   9.949226467E-06   2.421815563E-02        14
2,2     5.167323251E-03   1.058823425E-02    13     28   120    0   1.475958743E-05   3.014832252E-03         4
 1      3.582037868E-03   8.935547659E-03     8     10    10  180   8.762914758E-06   3.547830854E-03        22
 2      1.893918235E-02   1.071243767E-01    42     47    16  192   5.191475504E-06   2.169485013E-02        21
 3      2.667822681E-04   1.934196508E-04     2      4    10  180   7.500727127E-06   2.592815409E-04        21
 4      3.534143314E-03   6.776638209E-04     6     10    14  188   1.742820590E-05   7.557599221E-04        21
 5      6.773232908E-03   8.542432478E-03    13     16    16  192   1.666137224E-06   5.712574392E-03        22
 6      1.628272563E-03   2.001687407E-04     3      4     6  172   1.897017634E-05   7.710935717E-04        21
 7      6.538947245E-03   8.544854749E-02    16     22    12  144   6.701274447E-06   3.575398584E-02        22
 8      3.048817591E-04   1.020231700E-04     1      2    12  144   1.434254408E-05   3.048817591E-04        21
 9      1.323706113E-02   1.424869490E-01    27     30    10  140   2.324616131E-05   4.356237727E-02        22
10      7.628422585E-03   9.115952115E-03    15     17    14  109   2.078262261E-05   4.216982389E-03        21
11      1.255279721E-02   2.599464241E-02    28     29    16  112   2.735282320E-06   1.130035339E-02        22
12      1.060934378E-02   2.115944141E-02    21     22    12  104   2.913392366E-05   1.211067782E-02        21
13      4.546136707E-03   1.897835484E-02    10     12    10  100   4.021629743E-05   9.956930009E-03        21
14      2.807154436E-03   1.405517163E-02     8     12    14  108   2.626786119E-05   9.668068156E-03        22
15      3.449501665E-03   5.576976981E-04     6      9    10  100   1.515569256E-05   7.757943373E-04        22
16      1.771158559E-03   4.557122646E-05     3      1    10  100   4.557122646E-05   7.731939415E-04        21
17      2.558754842E-03   1.275066810E-03     5      7    12  104   2.331378027E-05   6.993266895E-04        22
18      1.193137800E-02   2.202933566E-02    24     29    12  104   1.328989686E-05   4.579528404E-03        22
19      1.393875635E-02   1.784939067E-02    25     25    14  108   2.145271718E-05   7.869506499E-03        21
20      9.748129331E-05   3.488237313E-04     1      3    14  108   5.514775416E-05   2.096708458E-04        22


SIMULATION RUN TIME (SECONDS):  5.000000000E+00
TOTAL BUS COLLISIONS:       715
TOTAL PACKETS TX:     2933

 AVG BUSBUSY            AVG USAGE            AVG IDLE
 7.254359865E-04      7.126278979E-04      9.793032884E-04

 TOT BUSBUSY            TOT USAGE            TOT IDLE
 2.127703455E+00      2.090137595E+00      2.872296545E+00


S = SIMULATED THROUGHPUT:  4.180275191E-01

G = AGGREGATE OFFERED LOAD AS A PERCENT OF BUS CAPACITY:  5.166694350E-01

E = EFFICIENCY (S/G):  8.090811856E-01

T = THEORETICAL THROUGHPUT:  3.406605438E-01
```

```
                ETHERNET SIMULATION PARAMETERS  RUN 5


ETHERNET BUS IO RATE:  1.000000000E+07

NUMBER OF MAIN FRAMES:          2
NUMBER OF HOSTS:        20

REFRESHES TO HOSTS IN SECONDS:  2.000000000E+00
DUMPS TO HOSTS IN SECONDS:  1.000000000E+01
OPERATOR REFRESH REQUESTS IN SECONDS:  1.000000000E+01


MAIN FRAME NUMBER:          1

I/O RATE OF MAIN FRAME:  1.000000000E+07
DISTANCE FROM REFRENCE POINT:  4.000000000E+02
FIRST REFRESH TIME:  1.000000000E-01 TO HOST:  1
FIRST DUMP TIME:  5.000000000E-01 TO HOST:  1

REFRESH OPERATION:
     BITS/PACKET:  7.680000000E+03
     SLTIME:  7.680000000E-04
     NUMBER OF PACKETS SENT TO EACH TERMINAL:         2

DUMP OPERATION:
     BITS/PACKET:  7.680000000E+03
     SLTIME:  7.680000000E-04
     NUMBER OF PACKETS SENT TO EACH TERMINAL:         3

MAIN FRAME NUMBER:          2

I/O RATE OF MAIN FRAME:  1.000000000E+07
DISTANCE FROM REFRENCE POINT:  6.750000000E+02
FIRST REFRESH TIME:  1.000000000E-01 TO HOST:  1
FIRST DUMP TIME:  5.000000000E-01 TO HOST:  1

REFRESH OPERATION:
     BITS/PACKET:  7.680000000E+03
     SLTIME:  7.680000000E-04
     NUMBER OF PACKETS SENT TO EACH TERMINAL:         2

DUMP OPERATION:
     BITS/PACKET:  7.680000000E+03
     SLTIME:  7.680000000E-04
     NUMBER OF PACKETS SENT TO EACH TERMINAL:         3


HOST NUMBER:          1
```

```
NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  5.000000000E+01
SLOT TIME:  1.024000000E-04


HOST NUMBER:         2

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  1.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:         3

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  1.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:         4

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  2.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:         5

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  2.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:         6

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  3.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:         7
```

```
NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  3.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        8

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  4.250000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        9

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  4.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:       10

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  5.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:       11

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  5.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:       12

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  6.000000000E+02
SLOT TIME:  1.024000000E-04
```

```
HOST NUMBER:       13

NUMBER OF TERMINALS:       10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  6.500000000E+02
SLOT TIME:  1.024000000E-04

HOST NUMBER:       14

NUMBER OF TERMINALS:       10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  7.000000000E+02
SLOT TIME:  1.024000000E-04

HOST NUMBER:       15

NUMBER OF TERMINALS:       10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  7.500000000E+02
SLOT TIME:  1.024000000E-04

HOST NUMBER:       16

NUMBER OF TERMINALS:       10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  8.000000000E+02
SLOT TIME:  1.024000000E-04

HOST NUMBER:       17

NUMBER OF TERMINALS:       10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  8.250000000E+02
SLOT TIME:  1.024000000E-04

HOST NUMBER:       18

NUMBER OF TERMINALS:       10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  8.750000000E+02
SLOT TIME:  1.024000000E-04
```

```
HOST NUMBER:        19

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  9.250000000E+02
SLOT TIME:  1.024000000E-04

HOST NUMBER:        20

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  9.750000000E+02
SLOT TIME:  1.024000000E-04
```

```
SOURCE  WAIT TIME         WAIT TIME          DEFER COLL  PKTS  PKTS  MINIMUM PKT        MAXIMUM PKT       MAX WAIT
        DEFER             COLLISION          COUNT COUNT TX     RX   WAIT TIME          WAIT TIME         RECEIVER
1,1     1.810647658E-01   1.657204155E-01    304   654   1268  144   1.837918887E-06    1.164544271E-02       7
1,2     2.266143831E-03   3.091825728E-03     16    51    270    0   1.348655303E-06    6.598557140E-04       6
2,1     1.551645668E-01   1.690158100E-01    269   654   1268  144   7.340739027E-07    2.061969289E-02       6
2,2     1.218650126E-02   5.908813370E-02     22    46    271    0   1.287314034E-05    1.983350825E-02       2
 1      1.501513449E-02   1.567904445E-01     35    38     16  212   7.339213977E-06    3.356108361E-02      22
 2      1.085233321E-02   1.326814345E-02     20    26     16  212   1.295918780E-05    3.952469603E-03      21
 3      6.616849313E-03   5.138509210E-03     15    18     14  208   1.029578255E-05    2.448943486E-03      22
 4      1.395083412E-02   2.370150259E-02     32    38     16  212   1.294628068E-05    5.380178801E-03      21
 5      6.224782409E-03   1.631170764E-02     16    21     18  216   1.281828508E-05    9.566719712E-03      22
 6      2.180582536E-02   7.921863467E-02     47    50     16  212   1.469411965E-05    1.757009696E-02      22
 7      4.495660527E-03   2.160222961E-03     11    14     12  204   1.960993918E-05    1.158307349E-03      22
 8      5.622237650E-03   1.118865496E-03     10    13     12  204   1.291623577E-05    7.730959762E-04      21
 9      7.501784856E-03   1.093887295E-02     14    17     14  208   1.407350237E-05    4.655029882E-03      22
10      9.991031189E-03   3.966216656E-02     21    23     12  105   1.229323853E-05    1.074755760E-02      22
11      1.996212168E-02   7.495155248E-02     40    48     16  112   1.281828508E-05    3.157838437E-02      21
12      1.119497983E-02   3.357650832E-02     23    29     14  108   2.151097169E-05    1.007852538E-02      21
13      5.462570416E-03   1.762393312E-02     15    19     14  108   1.326494309E-05    7.423179486E-03      22
14      1.039948583E-02   6.617864602E-02     22    23     14  108   7.747239289E-06    2.622394772E-02      22
15      1.154449150E-02   3.528242689E-02     21    24     12  104   2.118568324E-05    1.835051756E-02      22
16      1.657042562E-02   1.287164375E-01     36    39     14  108   1.554613289E-05    4.678715144E-02      21
17      1.091652503E-02   8.894349999E-03     18    18     12  104   2.770834567E-05    2.686883594E-03      21
18      9.050223157E-03   1.487979692E-03     17    19     18  116   1.684767239E-05    7.762251427E-04      22
19      2.553632499E-03   1.160871960E-03      8    13     18  116   1.856919519E-05    7.293750830E-04      22
20      4.155604273E-03   5.766554870E-04      7     8     10  100   2.118568324E-05    7.642700909E-04      21


SIMULATION RUN TIME (SECONDS):  5.000000000E+00
TOTAL BUS COLLISIONS:       903
TOTAL PACKETS TX:      3365


 AVG BUSBUSY            AVG USAGE            AVG IDLE
 7.241639341E-04      7.110333416E-04      7.617201669E-04

 TOT BUSBUSY            TOT USAGE            TOT IDLE
 2.436811638E+00      2.392627194E+00      2.563188362E+00


S = SIMULATED THROUGHPUT:  4.785254389E-01

G = AGGREGATE OFFERED LOAD AS A PERCENT OF BUS CAPACITY:  5.960601536E-01

E = EFFICIENCY (S/G):  8.028140045E-01

T = THEORETICAL THROUGHPUT:  3.734571980E-01
```

```
           ETHERNET SIMULATION PARAMETERS  RUN 6


ETHERNET BUS IO RATE:  1.000000000E+07

NUMBER OF MAIN FRAMES:        2
NUMBER OF HOSTS:       20

REFRESHES TO HOSTS IN SECONDS:  2.000000000E+00
DUMPS TO HOSTS IN SECONDS:  1.500000000E+01
OPERATOR REFRESH REQUESTS IN SECONDS:  1.800000000E+01


MAIN FRAME NUMBER:        1

I/O RATE OF MAIN FRAME:  1.000000000E+07
DISTANCE FROM REFRENCE POINT:  4.000000000E+02
FIRST REFRESH TIME:  1.000000000E-01 TO HOST:  1
FIRST DUMP TIME:  7.500000000E-01 TO HOST:  1

REFRESH OPERATION:
     BITS/PACKET:  7.680000000E+03
     SLTIME:  7.680000000E-04
     NUMBER OF PACKETS SENT TO EACH TERMINAL:        3

DUMP OPERATION:
     BITS/PACKET:  7.680000000E+03
     SLTIME:  7.680000000E-04
     NUMBER OF PACKETS SENT TO EACH TERMINAL:        3

MAIN FRAME NUMBER:        2

I/O RATE OF MAIN FRAME:  1.000000000E+07
DISTANCE FROM REFRENCE POINT:  6.750000000E+02
FIRST REFRESH TIME:  1.000000000E-01 TO HOST:  1
FIRST DUMP TIME:  7.500000000E-01 TO HOST:  1

REFRESH OPERATION:
     BITS/PACKET:  7.680000000E+03
     SLTIME:  7.680000000E-04
     NUMBER OF PACKETS SENT TO EACH TERMINAL:        3

DUMP OPERATION:
     BITS/PACKET:  7.680000000E+03
     SLTIME:  7.680000000E-04
     NUMBER OF PACKETS SENT TO EACH TERMINAL:        3

HOST NUMBER:         1

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  5.000000000E+01
SLOT TIME:  1.024000000E-04


HOST NUMBER:         2

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  1.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:         3

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  1.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:         4

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  2.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:         5

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  2.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:         6

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
```

```
DISTANCE FROM REFERENCE POINT:  3.000000000E+02
SLOT TIME:  1.024000000E-04

HOST NUMBER:        7

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  3.500000000E+02
SLOT TIME:  1.024000000E-04

HOST NUMBER:        8

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  4.250000000E+02
SLOT TIME:  1.024000000E-04

HOST NUMBER:        9

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  4.500000000E+02
SLOT TIME:  1.024000000E-04

HOST NUMBER:       10

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  5.000000000E+02
SLOT TIME:  1.024000000E-04

HOST NUMBER:       11

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  5.500000000E+02
SLOT TIME:  1.024000000E-04

HOST NUMBER:       12

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
```

```
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  6.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        13

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  6.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        14

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  7.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        15

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  7.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        16

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  8.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        17

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  8.250000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        18

NUMBER OF TERMINALS:        10
```

```
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  8.750000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:       19

NUMBER OF TERMINALS:       10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  9.250000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:       20

NUMBER OF TERMINALS:       10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  9.750000000E+02
SLOT TIME:  1.024000000E-04
```

```
SOURCE  WAIT TIME         WAIT TIME         DEFER COLL  PKTS  PKTS  MINIMUM PKT       MAXIMUM PKT       MAX WAIT
        DEFER             COLLISION         COUNT COUNT TX    RX    WAIT TIME         WAIT TIME         RECEIVER
1,1     2.370307275E-01   5.074086553E-01   393   750  1767   109   1.281828508E-05   3.899109578E-02          5
1,2     1.206537455E-02   1.177348447E-01    26    73   180     0   1.281828508E-05   4.123419450E-02          3
2,1     1.646572028E-01   3.317428928E-01   290   775  1768   109   7.887929485E-06   4.504764296E-02          4
2,2     1.444411272E-02   4.352973607E-02    31   104   180     0   6.533870135E-06   1.951154787E-02          5
 1      1.089847362E-02   9.916352111E-02    20    23    14   282   1.281828508E-05   3.356108361E-02         22
 2      9.698937420E-04   1.882024429E-04     3     4    12   276   1.504297039E-05   6.010467511E-04         21
 3      1.982146652E-02   1.654129087E-01    42    50    14   282   1.173998238E-05   4.069209048E-02         22
 4      1.498917246E-02   1.160161115E-01    29    33     8   264   2.251622156E-05   2.526172145E-02         22
 5      1.246141331E-02   1.394194987E-01    22    26    14   282   1.326494309E-05   4.189431343E-02         22
 6      9.012277933E-03   8.843881056E-02    22    27    10   270   1.381934616E-06   2.148231520E-02         22
 7      1.057395438E-02   3.208585466E-02    22    23    14   222   2.037647861E-05   1.017300840E-02         22
 8      1.941264170E-02   2.020484707E-01    39    42    10   210   1.287314034E-05   4.652743956E-02         22
 9      8.384591041E-03   1.469430272E-01    18    23    10   151   8.209818838E-06   4.218561795E-02         21
10      5.296128465E-03   1.270535799E-02     9    10     8   144   3.330338812E-05   8.409470898E-03         21
11      6.569530501E-03   1.214431386E-02    16    19    10   150   1.658823931E-05   4.135661190E-03         22
12      9.129451131E-03   2.205325982E-02    20    23    14   162   2.809005951E-05   7.901796094E-03         21
13      1.140520614E-02   1.087141523E-01    26    31    10   150   1.572334616E-06   4.419164806E-02         21
14      9.304660326E-03   1.827863829E-02    24    27    12   156   1.684767239E-05   4.313135260E-03         21
15      1.202115292E-02   1.315761853E-01    23    27    14   162   1.117010673E-05   3.819967885E-02         22
16      4.957126063E-03   1.264255237E-03     9    11     8   144   2.177188155E-05   7.734487860E-04         21
17      5.963658802E-03   1.613588091E-01    20    25     8   144   2.507811165E-05   4.066323361E-02         22
18      1.086509079E-02   4.502601711E-02    25    28     8   144   1.295918780E-05   1.619473276E-02         22
19      6.904378447E-03   1.417474472E-02    14    15    12   156   4.268339302E-06   1.071023825E-02         22
20      3.622291295E-04   1.476261430E-04     2     4     8   144   1.505774187E-05   2.049839274E-04         21


SIMULATION RUN TIME (SECONDS):  5.000000000E+00
TOTAL BUS COLLISIONS:      1022
TOTAL PACKETS TX:      4113

 AVG BUSBUSY            AVG USAGE            AVG IDLE
 7.455926372E-04      7.327214182E-04      4.700650336E-04

 TOT BUSBUSY            TOT USAGE            TOT IDLE
 3.066622517E+00      3.013683193E+00      1.933377483E+00


S = SIMULATED THROUGHPUT:  6.027366356E-01

G = AGGREGATE OFFERED LOAD AS A PERCENT OF BUS CAPACITY:  7.394815902E-01

E = EFFICIENCY (S/G):  8.150799754E-01

T = THEORETICAL THROUGHPUT:  4.251160767E-01
```

```
                ETHERNET SIMULATION PARAMETERS  RUN 7


ETHERNET BUS IO RATE:  1.000000000E+07

NUMBER OF MAIN FRAMES:        2
NUMBER OF HOSTS:       20

REFRESHES TO HOSTS IN SECONDS:  2.000000000E+00
DUMPS TO HOSTS IN SECONDS:  1.000000000E+01
OPERATOR REFRESH REQUESTS IN SECONDS:  1.000000000E+01


MAIN FRAME NUMBER:        1

I/O RATE OF MAIN FRAME:  1.000000000E+07
DISTANCE FROM REFRENCE POINT:  4.000000000E+02
FIRST REFRESH TIME:  1.000000000E-01 TO HOST:  1
FIRST DUMP TIME:  5.000000000E-01 TO HOST:  1

REFRESH OPERATION:
     BITS/PACKET:  7.680000000E+03
     SLTIME:  7.680000000E-04
     NUMBER OF PACKETS SENT TO EACH TERMINAL:        3

DUMP OPERATION:
     BITS/PACKET:  7.680000000E+03
     SLTIME:  7.680000000E-04
     NUMBER OF PACKETS SENT TO EACH TERMINAL:        3

MAIN FRAME NUMBER:        2

I/O RATE OF MAIN FRAME:  1.000000000E+07
DISTANCE FROM REFRENCE POINT:  6.750000000E+02
FIRST REFRESH TIME:  1.000000000E-01 TO HOST:  1
FIRST DUMP TIME:  5.000000000E-01 TO HOST:  1

REFRESH OPERATION:
     BITS/PACKET:  7.680000000E+03
     SLTIME:  7.680000000E-04
     NUMBER OF PACKETS SENT TO EACH TERMINAL:        3

DUMP OPERATION:
     BITS/PACKET:  7.680000000E+03
     SLTIME:  7.680000000E-04
     NUMBER OF PACKETS SENT TO EACH TERMINAL:        3


HOST NUMBER:        1
```

```
NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  5.000000000E+01
SLOT TIME:  1.024000000E-04


HOST NUMBER:        2

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  1.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        3

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  1.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        4

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  2.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        5

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  2.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        6

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  3.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        7
```

```
NUMBER OF TERMINALS:       10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  3.500000000E+02
SLOT TIME:  1.024000000E-04

HOST NUMBER:        8

NUMBER OF TERMINALS:       10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  4.250000000E+02
SLOT TIME:  1.024000000E-04

HOST NUMBER:        9

NUMBER OF TERMINALS:       10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  4.500000000E+02
SLOT TIME:  1.024000000E-04

HOST NUMBER:       10

NUMBER OF TERMINALS:       10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  5.000000000E+02
SLOT TIME:  1.024000000E-04

HOST NUMBER:       11

NUMBER OF TERMINALS:       10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  5.500000000E+02
SLOT TIME:  1.024000000E-04

HOST NUMBER:       12

NUMBER OF TERMINALS:       10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  6.000000000E+02
SLOT TIME:  1.024000000E-04
```

```
HOST NUMBER:       13

NUMBER OF TERMINALS:       10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  6.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:       14

NUMBER OF TERMINALS:       10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  7.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:       15

NUMBER OF TERMINALS:       10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  7.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:       16

NUMBER OF TERMINALS:       10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  8.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:       17

NUMBER OF TERMINALS:       10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  8.250000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:       18

NUMBER OF TERMINALS:       10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  8.750000000E+02
SLOT TIME:  1.024000000E-04
```

```
HOST NUMBER:       19

NUMBER OF TERMINALS:       10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  9.250000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:       20

NUMBER OF TERMINALS:       10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  9.750000000E+02
SLOT TIME:  1.024000000E-04
```

```
SOURCE  WAIT TIME         WAIT TIME         DEFER COLL  PKTS  PKTS  MINIMUM PKT       MAXIMUM PKT       MAX WAIT
        DEFER             COLLISION         COUNT COUNT TX    RX    WAIT TIME         WAIT TIME         RECEIVER
1,1     2.719457622E-01   4.385093501E-01   491   1060  1886  159   3.492661449E-07   4.255523132E-02      10
1,2     1.462727162E-02   9.309791257E-02    36    101   270    0   9.949222829E-06   4.562059513E-02       1
2,1     2.958089247E-01   6.084829826E-01   506   1060  1887  159   5.700950772E-06   4.574975350E-02      19
2,2     1.873873317E-02   7.063426169E-02    38    106   270    0   1.286123711E-05   2.367406035E-02       7
 1      4.924634049E-03   1.783785421E-01    16     16    14  282   4.330977962E-06   4.470828158E-02      21
 2      1.543783198E-02   1.099194013E-01    30     36    16  288   1.897017634E-05   2.127921567E-02      22
 3      1.385705836E-02   1.578993556E-01    32     37    18  294   1.286123711E-05   4.357197726E-02      21
 4      1.158659699E-02   2.230447898E-02    21     23    14  282   1.304494843E-05   7.040972140E-03      22
 5      7.582902982E-03   1.077308013E-03    12     17    12  276   1.377979371E-05   7.742275436E-04      22
 6      1.437693147E-02   4.463141060E-02    32     34    14  282   1.953393059E-05   1.857997745E-02      21
 7      1.731720655E-02   1.891495735E-01    36     37    18  293   2.145271718E-05   4.406388480E-02      21
 8      1.546888316E-02   3.295379271E-02    33     35    16  228   1.511410295E-05   1.145871804E-02      22
 9      5.581492779E-03   2.818103175E-02    13     14    18  234   1.392392320E-05   1.865694809E-02      21
10      1.425543784E-02   1.221294051E-01    30     37    16  168   1.057525633E-05   3.575398584E-02      21
11      1.586802868E-02   2.167152521E-01    30     35    18  174   2.331378027E-05   4.585418590E-02      21
12      6.635561517E-03   3.174896410E-03    13     18    14  162   2.626786119E-05   1.152082102E-03      21
13      8.190187967E-03   2.685778737E-02    21     25    12  156   1.515569256E-05   1.103517119E-02      22
14      5.087411193E-03   3.203545696E-02    11     16    16  168   1.504784641E-05   1.789299291E-02      22
15      2.305310556E-02   3.398676075E-01    53     60    18  174   1.729569282E-05   4.574975351E-02      22
16      1.243480269E-02   1.415785901E-01    27     32    16  168   1.434254408E-05   4.471788158E-02      22
17      1.014703615E-02   1.883588612E-02    20     24    16  168   1.511410295E-05   1.087133744E-02      22
18      1.879622745E-02   2.393639242E-01    38     46    20  180   9.868579710E-06   5.041173651E-02      22
19      1.563797510E-02   2.172636672E-01    35     38    16  168   1.897017634E-05   4.432080644E-02      21
20      1.494335927E-02   8.005762772E-02    31     40    16  168   5.719332073E-06   2.170445011E-02      22


SIMULATION RUN TIME (SECONDS):  5.000000000E+00
TOTAL BUS COLLISIONS:     1372
TOTAL PACKETS TX:      4631

 AVG BUSBUSY           AVG USAGE            AVG IDLE
 7.357964826E-04      7.222947941E-04      3.438839320E-04

 TOT BUSBUSY           TOT USAGE            TOT IDLE
 3.407473511E+00      3.344947192E+00      1.592526489E+00


S = SIMULATED THROUGHPUT:  6.689894393E-01

G = AGGREGATE OFFERED LOAD AS A PERCENT OF BUS CAPACITY:  8.573644662E-01

E = EFFICIENCY (S/G):  7.802859399E-01

T = THEORETICAL THROUGHPUT:  4.616027066E-01
```

ETHERNET SIMULATION PARAMETERS RUN 8

```
ETHERNET BUS IO RATE:  1.000000000E+07

NUMBER OF MAIN FRAMES:          2
NUMBER OF HOSTS:          20

REFRESHES TO HOSTS IN SECONDS:  2.000000000E+00
DUMPS TO HOSTS IN SECONDS:  1.500000000E+01
OPERATOR REFRESH REQUESTS IN SECONDS:  1.800000000E+01


MAIN FRAME NUMBER:          1

I/O RATE OF MAIN FRAME:  1.000000000E+07
DISTANCE FROM REFRENCE POINT:  4.000000000E+02
FIRST REFRESH TIME:  1.000000000E-01 TO HOST:  1
FIRST DUMP TIME:  7.500000000E-01 TO HOST:  1

REFRESH OPERATION:
     BITS/PACKET:  7.680000000E+03
     SLTIME:  7.680000000E-04
     NUMBER OF PACKETS SENT TO EACH TERMINAL:        4

DUMP OPERATION:
     BITS/PACKET:  7.680000000E+03
     SLTIME:  7.680000000E-04
     NUMBER OF PACKETS SENT TO EACH TERMINAL:        3

MAIN FRAME NUMBER:          2

I/O RATE OF MAIN FRAME:  1.000000000E+07
DISTANCE FROM REFRENCE POINT:  6.750000000E+02
FIRST REFRESH TIME:  1.000000000E-01 TO HOST:  1
FIRST DUMP TIME:  7.500000000E-01 TO HOST:  1

REFRESH OPERATION:
     BITS/PACKET:  7.680000000E+03
     SLTIME:  7.680000000E-04
     NUMBER OF PACKETS SENT TO EACH TERMINAL:        4

DUMP OPERATION:
     BITS/PACKET:  7.680000000E+03
     SLTIME:  7.680000000E-04
     NUMBER OF PACKETS SENT TO EACH TERMINAL:        3


HOST NUMBER:          1
```

```
NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  5.000000000E+01
SLOT TIME:  1.024000000E-04


HOST NUMBER:         2

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  1.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:         3

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  1.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:         4

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  2.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:         5

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  2.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:         6

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  3.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:         7
```

```
NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  3.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:         8

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  4.250000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:         9

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  4.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        10

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  5.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        11

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  5.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        12

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  6.000000000E+02
SLOT TIME:  1.024000000E-04
```

```
HOST NUMBER:        13

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  6.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        14

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  7.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        15

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  7.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        16

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  8.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        17

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  8.250000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        18

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  8.750000000E+02
SLOT TIME:  1.024000000E-04
```

```
HOST NUMBER:        19

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  9.250000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        20

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  9.750000000E+02
SLOT TIME:  1.024000000E-04
```

```
SOURCE  WAIT TIME         WAIT TIME         DEFER COLL  PKTS  PKTS  MINIMUM PKT       MAXIMUM PKT       MAX WAIT
        DEFER             COLLISION         COUNT COUNT TX    RX    WAIT TIME         WAIT TIME         RECEIVER
1,1     2.733978410E-01   1.123510230E+00   484  1061  2181   105   5.539656001E-06   5.227487736E-02          1
1,2     8.013151655E-03   3.446938634E-02    15    46   180     0   1.261828508E-05   2.035724523E-02          4
2,1     2.629496621E-01   8.389442670E-01   453  1045  2260   105   3.492558690E-07   4.504764297E-02         13
2,2     7.561185768E-03   2.303339585E-02    15    51   180     0   1.311837559E-05   1.102233979E-02          6
 1      1.340990410E-02   1.051901137E-01    26    28    14   356   3.603338545E-06   4.652743956E-02         22
 2      8.403249868E-03   7.857302352E-02    19    23    10   340   1.856919519E-05   2.527132144E-02         21
 3      1.157479593E-02   2.599909508E-01    25    31     8   332   1.343675118E-05   4.503804298E-02         22
 4      1.286764293E-02   1.083252374E-01    31    37    12   348   1.837340919E-06   3.819007885E-02         22
 5      3.257097500E-03   7.371733573E-04     6    10     8   293   1.776583641E-05   7.746974071E-04         22
 6      2.636311731E-02   3.653529295E-01    56    59    12   308   5.327295788E-06   5.216904985E-02         21
 7      9.326468929E-03   4.465141501E-02    21    26    10   200   2.212390689E-05   1.043222465E-02         22
 8      1.945635520E-02   2.400839698E-01    44    52    14   216   1.750650909E-05   3.959290481E-02         22
 9      1.666864255E-02   1.523513190E-01    29    31    10   200   1.465977238E-05   4.504764297E-02         21
10      1.994249636E-03   2.904197767E-04     3     4     8   192   1.407350237E-05   7.159617014E-04         22
11      6.674564941E-03   3.655496954E-02    18    21     8   192   2.710650579E-05   2.035724524E-02         22
12      9.914512510E-03   6.302115156E-02    21    24    12   206   9.621174927E-06   2.950643563E-02         22
13      2.514484572E-04   7.696654899E-05     2     2    12   208   2.916918097E-05   2.118348632E-04         21
14      1.162550435E-02   9.999000497E-02    23    27    10   200   1.610567346E-05   4.204272869E-02         22
15      8.824339598E-03   1.018965256E-01    18    21    10   200   1.397024542E-05   4.281215298E-02         22
16      1.755091543E-02   3.064768856E-01    38    44    12   208   1.467160533E-05   4.281215298E-02         22
17      7.682724272E-03   8.010480571E-02    21    25    10   200   1.792413488E-05   1.910143942E-02         22
18      6.763993817E-03   8.316929056E-02    17    22    10   200   2.221301818E-05   3.960250480E-02         22
19      1.263456016E-02   1.780559250E-01    28    28    12   208   1.291623577E-05   4.265413771E-02         22
20      4.479519009E-03   8.058956520E-03     9    14     8   192   1.504297039E-05   5.096511311E-03         22


SIMULATION RUN TIME (SECONDS):  5.000000000E+00
TOTAL BUS COLLISIONS:     1301
TOTAL PACKETS TX:      5011

 AVG BUSBUSY            AVG USAGE            AVG IDLE
 7.531220408E-04       7.401061646E-04      2.446827886E-04

 TOT BUSBUSY            TOT USAGE            TOT IDLE
 3.773894547E+00       3.708671991E+00      1.226105453E+00


S = SIMULATED THROUGHPUT:  7.417343982E-01

G = AGGREGATE OFFERED LOAD AS A PERCENT OF BUS CAPACITY:  9.174732644E-01

E = EFFICIENCY (S/G):  8.084534198E-01

T = THEORETICAL THROUGHPUT:  4.784803417E-01
```

```
                ETHERNET SIMULATION PARAMETERS  RUN 9


ETHERNET BUS IO RATE:  1.000000000E+07

NUMBER OF MAIN FRAMES:          2
NUMBER OF HOSTS:         20

REFRESHES TO HOSTS IN SECONDS:  2.000000000E+00
DUMPS TO HOSTS IN SECONDS:  1.500000000E+01
OPERATOR REFRESH REQUESTS IN SECONDS:  1.800000000E+01


MAIN FRAME NUMBER:          1

I/O RATE OF MAIN FRAME:  1.000000000E+07
DISTANCE FROM REFRENCE POINT:  4.000000000E+02
FIRST REFRESH TIME:  1.000000000E-01 TO HOST:  1
FIRST DUMP TIME:  7.500000000E-01 TO HOST:  1

REFRESH OPERATION:
     BITS/PACKET:  7.680000000E+03
     SLTIME:  7.680000000E-04
     NUMBER OF PACKETS SENT TO EACH TERMINAL:         4

DUMP OPERATION:
     BITS/PACKET:  7.680000000E+03
     SLTIME:  7.680000000E-04
     NUMBER OF PACKETS SENT TO EACH TERMINAL:         5

MAIN FRAME NUMBER:          2

I/O RATE OF MAIN FRAME:  1.000000000E+07
DISTANCE FROM REFRENCE POINT:  6.750000000E+02
FIRST REFRESH TIME:  1.000000000E-01 TO HOST:  1
FIRST DUMP TIME:  7.500000000E-01 TO HOST:  1

REFRESH OPERATION:
     BITS/PACKET:  7.680000000E+03
     SLTIME:  7.680000000E-04
     NUMBER OF PACKETS SENT TO EACH TERMINAL:         4

DUMP OPERATION:
     BITS/PACKET:  7.680000000E+03
     SLTIME:  7.680000000E-04
     NUMBER OF PACKETS SENT TO EACH TERMINAL:         5


HOST NUMBER:          1
```

```
NUMBER OF TERMINALS:         10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  5.000000000E+01
SLOT TIME:  1.024000000E-04


HOST NUMBER:         2

NUMBER OF TERMINALS:         10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  1.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:         3

NUMBER OF TERMINALS:         10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  1.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:         4

NUMBER OF TERMINALS:         10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  2.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:         5

NUMBER OF TERMINALS:         10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  2.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:         6

NUMBER OF TERMINALS:         10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  3.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:         7
```

```
NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  3.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        8

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  4.250000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        9

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  4.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:       10

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  5.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:       11

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  5.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:       12

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  6.000000000E+02
SLOT TIME:  1.024000000E-04
```

```
HOST NUMBER:        13

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  6.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        14

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  7.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        15

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  7.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        16

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  8.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        17

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  8.250000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        18

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  8.750000000E+02
SLOT TIME:  1.024000000E-04
```

```
HOST NUMBER:        19

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  9.250000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        20

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  9.750000000E+02
SLOT TIME:  1.024000000E-04
```

```
SOURCE  WAIT TIME        WAIT TIME        DEFER COLL  PKTS  PKTS  MINIMUM PKT      MAXIMUM PKT      MAX WAIT
        DEFER            COLLISION        COUNT COUNT TX    RX    WAIT TIME        WAIT TIME        RECEIVER
1,1     2.703256931E-01  1.176903209E+00  499  1161  2220  125   5.246946046E-06  5.118529168E-02         2
1,2     1.865619707E-02  1.167484597E-01   37    89   300    0   1.377979371E-05  3.089139983E-02         2
2,1     2.998174267E-01  7.799498859E-01  538  1252  2298  125   3.492661449E-07  4.407348478E-02         9
2,2     3.223942681E-03  2.880298278E-03   13    53   300    0   1.802308066E-05  7.652196503E-04         2
 1      1.497268106E-02  1.911889414E-01   37    38    14  396   1.186238178E-06  4.574975350E-02        22
 2      1.589314638E-02  1.802911825E-01   40    49    14  396   1.504784641E-05  4.407348479E-02        22
 3      1.436212680E-02  5.578967428E-02   29    32    12  388   1.953393059E-05  8.910629642E-03        22
 4      1.125215642E-02  6.670501858E-02   23    25    10  340   1.742820590E-05  2.293349295E-02        22
 5      1.856476285E-02  3.540173950E-01   40    41     8  332   1.286123711E-05  4.471894562E-02        22
 6      1.157075918E-02  3.457559238E-02   24    29    16  322   1.906386677E-05  2.290463607E-02        22
 7      1.303987414E-02  3.507527231E-01   31    35    12  208   1.450990727E-06  4.562059512E-02        22
 8      2.395732977E-02  2.938892864E-01   50    56    12  208   1.377979371E-05  3.976052006E-02        21
 9      1.815386134E-02  3.174317819E-01   40    46    12  208   2.753327972E-05  5.119489166E-02        21
10      1.912276672E-02  1.844309243E-01   40    44    14  216   8.663140514E-06  3.576358584E-02        21
11      2.334521569E-02  1.466372555E-01   45    46    12  208   2.991670361E-05  4.580607213E-02        22
12      1.770802871E-02  3.349063250E-01   39    43    14  216   2.546823184E-06  4.731612073E-02        22
13      4.558952800E-03  1.664202812E-03    9    11     6  184   1.679131131E-05  7.744477618E-04        22
14      1.802069152E-02  3.527038150E-01   42    46    16  224   5.569123170E-06  5.119489167E-02        22
15      1.056047927E-02  1.901264079E-01   25    31    14  216   2.675251990E-06  4.736010360E-02        22
16      1.497411162E-02  1.756108735E-01   30    33    10  200   2.352390899E-05  5.118529168E-02        22
17      2.082819730E-02  3.647347991E-01   45    51    12  208   1.291358799E-05  4.586378589E-02        21
18      8.364229629E-03  3.369284320E-03   17    21    16  224   7.635948438E-08  1.380894911E-03        21
19      1.283780342E-02  5.146855616E-02   26    30    14  216   1.742820590E-05  1.724789856E-02        22
20      1.968865010E-02  3.263923620E-01   47    54    12  208   9.647694349E-06  4.736970358E-02        22


SIMULATION RUN TIME (SECONDS):  5.000000000E+00
TOTAL BUS COLLISIONS:     1568
TOTAL PACKETS TX:     5368

 AVG BUSBUSY           AVG USAGE          AVG IDLE
 7.504433645E-04     7.370014885E-04     1.810022391E-04

 TOT BUSBUSY           TOT USAGE          TOT IDLE
 4.028379981E+00     3.956223990E+00     9.716200193E-01


S = SIMULATED THROUGHPUT:  7.912447981E-01

G = AGGREGATE OFFERED LOAD AS A PERCENT OF BUS CAPACITY:  9.964134216E-01

E = EFFICIENCY (S/G):  7.940928744E-01

T = THEORETICAL THROUGHPUT:  4.991017436E-01
```

```
                ETHERNET SIMULATION PARAMETERS  RUN 10


ETHERNET BUS IO RATE:  1.000000000E+07

NUMBER OF MAIN FRAMES:         2
NUMBER OF HOSTS:         20

REFRESHES TO HOSTS IN SECONDS:  2.000000000E+00
DUMPS TO HOSTS IN SECONDS:  1.500000000E+01
OPERATOR REFRESH REQUESTS IN SECONDS:  1.800000000E+01


MAIN FRAME NUMBER:         1

I/O RATE OF MAIN FRAME:  1.000000000E+07
DISTANCE FROM REFRENCE POINT:  4.000000000E+02
FIRST REFRESH TIME:  1.000000000E-01 TO HOST:  1
FIRST DUMP TIME:  7.500000000E-01 TO HOST:  1

REFRESH OPERATION:
     BITS/PACKET:  7.680000000E+03
     SLTIME:  7.680000000E-04
     NUMBER OF PACKETS SENT TO EACH TERMINAL:         5

DUMP OPERATION:
     BITS/PACKET:  7.680000000E+03
     SLTIME:  7.680000000E-04
     NUMBER OF PACKETS SENT TO EACH TERMINAL:         5

MAIN FRAME NUMBER:         2

I/O RATE OF MAIN FRAME:  1.000000000E+07
DISTANCE FROM REFRENCE POINT:  6.750000000E+02
FIRST REFRESH TIME:  1.000000000E-01 TO HOST:  1
FIRST DUMP TIME:  7.500000000E-01 TO HOST:  1

REFRESH OPERATION:
     BITS/PACKET:  7.680000000E+03
     SLTIME:  7.680000000E-04
     NUMBER OF PACKETS SENT TO EACH TERMINAL:         5

DUMP OPERATION:
     BITS/PACKET:  7.680000000E+03
     SLTIME:  7.680000000E-04
     NUMBER OF PACKETS SENT TO EACH TERMINAL:         5


HOST NUMBER:         1
```

```
NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  5.000000000E+01
SLOT TIME:  1.024000000E-04


HOST NUMBER:         2

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  1.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:         3

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  1.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:         4

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  2.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:         5

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  2.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:         6

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  3.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBEP:         7
```

```
NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  3.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:         8

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  4.250000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:         9

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  4.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        10

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  5.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        11

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  5.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        12

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  6.000000000E+02
SLOT TIME:  1.024000000E-04
```

```
HOST NUMBER:       13

NUMBER OF TERMINALS:       10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  6.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:       14

NUMBER OF TERMINALS:       10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  7.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:       15

NUMBER OF TERMINALS:       10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  7.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:       16

NUMBER OF TERMINALS:       10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  8.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:       17

NUMBER OF TERMINALS:       10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  8.250000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:       18

NUMBER OF TERMINALS:       10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  8.750000000E+02
SLOT TIME:  1.024000000E-04
```

```
HOST NUMBER:        19

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  9.250000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        20

NUMBER OF TERMINALS:        10
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  9.750000000E+02
SLOT TIME:  1.024000000E-04
```

```
SOURCE  WAIT TIME         WAIT TIME          DEFER COLL   PKTS  PKTS  MINIMUM PKT        MAXIMUM PKT        MAX WAIT
        DEFER             COLLISION          COUNT COUNT  TX    RX    WAIT TIME          WAIT TIME          RECEIVER
1,1     3.797643877E-01   1.578963119E+00    638   1413   2414  104   1.866317859E-06    4.679675144E-02        4
1,2     6.161507162E-03   7.951978407E-02     15     89    300    0   6.312364519E-06    4.679675142E-02        6
2,1     3.189756690E-01   1.355480172E+00    587   1387   2520  104   3.492661449E-07    5.228447735E-02        5
2,2     1.636613727E-02   1.206990305E-01     37    106    300    0   1.287314034E-05    3.819967884E-02        5
 1      2.066834419E-02   2.609943531E-01     38     42     10  350   1.338512271E-05    5.216904985E-02       22
 2      1.397258936E-02   8.804660182E-02     29     33     10  350   5.356003095E-06    2.095835671E-02       22
 3      7.830172115E-03   1.076674705E-01     22     27     10  350   1.222321582E-05    4.503804298E-02       22
 4      1.795728749E-02   9.156597235E-02     33     38     14  370   1.505774187E-05    4.069209048E-02       22
 5      1.062808597E-02   9.327197933E-02     26     32     12  360   1.303247155E-05    3.157838438E-02       22
 6      1.546283708E-02   8.342022405E-02     28     33      8  340   1.291623577E-05    2.405577146E-02       22
 7      1.968209737E-02   3.637314673E-01     41     44      8  240   2.465773493E-05    4.265413772E-02       22
 8      1.813111880E-02   2.407684820E-01     35     39     10  250   1.686538795E-05    4.280255298E-02       21
 9      2.011421333E-02   2.692444312E-01     43     47     12  260   1.465977238E-05    5.128559319E-02       21
10      1.549789622E-02   4.283524921E-02     28     35     10  250   1.357356517E-05    1.580039217E-02       22
11      1.479612446E-02   1.156006199E-01     36     41     12  260   1.283657017E-05    2.950643563E-02       21
12      1.522372255E-02   2.509058965E-01     38     42     10  250   5.720435824E-06    4.491848459E-02       22
13      1.017252432E-02   9.830431364E-02     24     28     10  250   2.092001172E-05    4.188471344E-02       21
14      1.826955813E-02   2.395713677E-01     35     41     10  250   1.500945490E-05    5.216904985E-02       22
15      1.987241699E-02   2.173808715E-01     39     40     10  250   1.520560060E-05    4.652743956E-02       22
16      1.772666134E-02   2.586698886E-01     36     37     10  250   2.641987836E-05    4.265413772E-02       21
17      5.868340765E-03   3.840572082E-03     11     13      6  230   1.343675118E-05    1.205339964E-03       21
18      7.235507945E-03   1.315503702E-01     18     22     10  244   1.311837559E-05    3.960250481E-02       22
19      1.652553546E-02   2.037754628E-01     33     35     12  210   1.291623577E-05    4.420124806E-02       22
20      1.326947744E-02   4.406479808E-02     27     28     14  220   1.295918780E-05    2.623354771E-02       21


SIMULATION RUN TIME (SECONDS):  5.000000000E+00
TOTAL BUS COLLISIONS:     1742
TOTAL PACKETS TX:     5742


 AVG BUSBUSY           AVG USAGE           AVG IDLE
 7.574802682E-04     7.438890961E-04     1.132964647E-04

 TOT BUSBUSY           TOT USAGE           TOT IDLE
 4.349451700E+00     4.271411190E+00     6.505483001E-01


S = SIMULATED THROUGHPUT:  8.542822379E-01

G = AGGREGATE OFFERED LOAD AS A PERCENT OF BUS CAPACITY:  1.090037739E+00

E = EFFICIENCY (S/G):  7.837180380E-01

T = THEORETICAL THROUGHPUT:  5.215397389E-01
```

```
                                         SIMULATION RUN PARAMETERS

S.T.    - SIMULATION RUN TIME
E.B.    - ETHERNET BUS
M.F.    - MAIN FRAME
T.F.D.  - TAPE FILE DUMP
H.      - HOSTS
T/H     - TERMINALS PER HOST
S.R.    - SCREEN REFRESH
O.R.    - OPERATOR REQUEST


                  I/O RATES                        NUMBER OF     NUMBER PKTS FOR  BITS/PKT FOR                  AVG TIMES (SEC)
RUN   S.T.        E.B.      M.F.      H.        M.F. H.  T/H   S.R. T.F.D. O.P. S.R.      T.F.D.    O.R.      S.R. T.F.D. O.R.
  1   5.0E+00   1.0E+07   1.0E+07   2.4E+06   2    20   10    2     2      1   7.7E+03   7.7E+03   1.0E+03    2.0E+00      1.5E+01      1.8E+02
  2   5.0E+00   1.0E+07   1.0E+07   2.4E+06   2    20   10    2     2      1   7.7E+03   7.7E+03   1.0E+03    2.0E+00      1.0E+01      1.0E+01
  3   5.0E+00   1.0E+07   1.0E+07   2.4E+06   2    20   10    2     2      1   7.7E+03   7.7E+03   1.0E+03    2.0E+00      4.0E+00      1.0E+01
  4   5.0E+00   1.0E+07   1.0E+07   2.4E+06   2    20   10    4     4      1   7.7E+03   7.7E+03   1.0E+03    2.0E+00      1.5E+01      1.8E+01
  5   5.0E+00   1.0E+07   1.0E+07   2.4E+06   2    20   10    4     6      1   7.7E+03   7.7E+03   1.0E+03    2.0E+00      1.0E+01      1.0E+01
  6   5.0E+00   1.0E+07   1.0E+07   2.4E+06   2    20   10    6     6      1   7.7E+03   7.7E+03   1.0E+03    2.0E+00      1.5E+01      1.8E+01
  7   5.0E+00   1.0E+07   1.0E+07   2.4E+06   2    20   10    6     6      1   7.7E+03   7.7E+03   1.0E+03    2.0E+00      1.0E+01      1.0E+01
  8   5.0E+00   1.0E+07   1.0E+07   2.4E+06   2    20   10    6     6      1   7.7E+03   7.7E+03   1.0E+03    2.0E+00      1.5E+01      1.8E+01
  9   5.0E+00   1.0E+07   1.0E+07   2.4E+06   2    20   10    8    10      1   7.7E+03   7.7E+03   1.0E+03    2.0E+00      1.5E+01      1.8E+01
 10   5.0E+00   1.0E+07   1.0E+07   2.4E+06   2    20   10   10    10      1   7.7E+03   7.7E+03   1.0E+03    2.0E+00      1.5E+01      1.8E+01


                                          SIMULATION RUN RESULTS

 G    - AGGREGRATE OFFERED LOAD AS A PERCENT OF BUS CAPACITY
 S    - SIMULATED THROUGHPUT
 T    - THEORETICAL THROUGHPUT
 E    - EFFICIENCY
 TOT WAIT DEFER TIME   - MAXIMUM TOTAL WAITING TIME OF A DEVICE DUE TO PACKETS BEING DEFFERED
 TOT WAIT COLL TIME    - MAXIMUM TOTAL WAITING TIME OF A DEVICE DUE TO PACKET COLLISIONS
 MIN WAIT PACKET       - MINIMUM PACKET WAIT TIME TO ACCESS ETHERNET BUS
 MAX WAIT PACKET       - MAXIMUM PACKET WAIT TIME TO ACCESS ETHERNET BUS


                                                NUM     PKTS     TOT WAIT       TOT WAIT        MIN WAIT     MAX WAIT
 RUN   G          S          T          E           COLS    TMIT     DEFER TIME     COLL TIME       PACKET       PACKET
   1   2.1E-01   1.8E-01   1.7E-01   8.7E-01    182     1217     5.1E-02        5.8E-02         9.9E-06      5.9E-03
   2   2.9E-01   2.3E-01   2.3E-01   8.0E-01    431     1797     8.3E-02        7.1E-02         1.7E-06      2.6E-02
   3   3.2E-01   2.8E-01   2.4E-01   8.6E-01    331     2077     7.7E-02        5.2E-02         1.1E-06      2.5E-02
   4   5.2E-01   4.2E-01   3.4E-01   8.1E-01    715     2933     1.8E-01        2.4E-01         1.7E-06      4.4E-02
   5   6.0E-01   4.8E-01   3.7E-01   8.0E-01    903     3365     1.8E-01        1.7E-01         7.3E-07      4.7E-02
   6   7.4E-01   6.0E-01   4.3E-01   8.2E-01   1022     4113     2.4E-01        5.1E-01         1.4E-06      4.7E-02
   7   8.6E-01   6.7E-01   4.6E-01   7.8E-01   1372     4631     3.0E-01        6.1E-01         3.5E-07      5.0E-02
   8   9.2E-01   7.4E-01   4.8E-01   8.1E-01   1301     5011     2.7E-01        1.1E+00         3.5E-07      5.2E-02
   9   1.0E+00   7.9E-01   5.0E-01   7.9E-01   1568     5368     3.0E-01        1.2E+00         7.6E-08      5.1E-02
  10   1.1E+00   8.5E-01   5.2E-01   7.8E-01   1742     5742     3.8E-01        1.6E+00         3.5E-07      5.2E-02
```

ETHERNET SIMULATION PARAMETERS RUN 11

```
ETHERNET BUS IO RATE:  1.000000000E+07

NUMBER OF MAIN FRAMES:        2
NUMBER OF HOSTS:       20

REFRESHES TO HOSTS IN SECONDS:  4.000000000E+01
DUMPS TO HOSTS IN SECONDS:  4.000000000E+01
OPERATOR REFRESH REQUESTS IN SECONDS:  1.000000000E+00


MAIN FRAME NUMBER:        1

I/O RATE OF MAIN FRAME:  1.000000000E+07
DISTANCE FROM REFRENCE POINT:  4.000000000E+02
FIRST REFRESH TIME:  2.000000000E+00 TO HOST:  1
FIRST DUMP TIME:  2.000000000E+00 TO HOST:  1

REFRESH OPERATION:
     BITS/PACKET:  7.680000000E+03
     SLTIME:  7.680000000E-04
     NUMBER OF PACKETS SENT TO EACH TERMINAL:        1

DUMP OPERATION:
     BITS/PACKET:  7.680000000E+03
     SLTIME:  7.680000000E-04
     NUMBER OF PACKETS SENT TO EACH TERMINAL:        1

MAIN FRAME NUMBER:        2

I/O RATE OF MAIN FRAME:  1.000000000E+07
DISTANCE FROM REFRENCE POINT:  6.750000000E+02
FIRST REFRESH TIME:  2.000000000E+00 TO HOST:  1
FIRST DUMP TIME:  2.000000000E+00 TO HOST:  1

REFRESH OPERATION:
     BITS/PACKET:  7.680000000E+03
     SLTIME:  7.680000000E-04
     NUMBER OF PACKETS SENT TO EACH TERMINAL:        1

DUMP OPERATION:
     BITS/PACKET:  7.680000000E+03
     SLTIME:  7.680000000E-04
     NUMBER OF PACKETS SENT TO EACH TERMINAL:        1


HOST NUMBER:        1
```

```
NUMBER OF TERMINALS:        15
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  5.000000000E+01
SLOT TIME:  1.024000000E-04


HOST NUMBER:          2

NUMBER OF TERMINALS:        15
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  1.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:          3

NUMBER OF TERMINALS:        15
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  1.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:          4

NUMBER OF TERMINALS:        15
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  2.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:          5

NUMBER OF TERMINALS:        15
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  2.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:          6

NUMBER OF TERMINALS:        15
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  3.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:          7
```

```
NUMBER OF TERMINALS:        15
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  3.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        8

NUMBER OF TERMINALS:        15
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  4.250000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:        9

NUMBER OF TERMINALS:        15
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  4.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:       10

NUMBER OF TERMINALS:        15
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  5.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:       11

NUMBER OF TERMINALS:        15
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  5.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:       12

NUMBER OF TERMINALS:        15
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  6.000000000E+02
SLOT TIME:  1.024000000E-04
```

```
HOST NUMBER:       13

NUMBER OF TERMINALS:       15
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  6.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:       14

NUMBER OF TERMINALS:       15
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  7.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:       15

NUMBER OF TERMINALS:       15
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  7.500000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:       16

NUMBER OF TERMINALS:       15
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  8.000000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:       17

NUMBER OF TERMINALS:       15
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  8.250000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:       18

NUMBER OF TERMINALS:       15
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  8.750000000E+02
SLOT TIME:  1.024000000E-04
```

```
HOST NUMBER:       19

NUMBER OF TERMINALS:       15
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  9.250000000E+02
SLOT TIME:  1.024000000E-04


HOST NUMBER:       20

NUMBER OF TERMINALS:       15
I/O RATE OF HOST:  2.400000000E+06
BITS PER PACKET:  1.024000000E+03
DISTANCE FROM REFERENCE POINT:  9.750000000E+02
SLOT TIME:  1.024000000E-04
```

```
SOURCE  WAIT TIME        WAIT TIME         DEFER COLL  PKTS  PKTS  MINIMUM PKT       MAXIMUM PKT       MAX WAIT
        DEFER            COLLISION         COUNT COUNT TX     RX   WAIT TIME         WAIT TIME         RECEIVER
1,1     6.953154716E-01  2.598335783E+00 1407  3093  1530 1501    2.004455528E-07   5.052296706E-02        17
1,2     7.646899686E-04  1.373708752E-03   11    30    30    0    1.731548373E-05   1.119029581E-04         2
2,1     6.243650919E-01  1.640677116E+00 1247  2856  1530 1500    2.906301451E-06   5.215944986E-02        16
2,2     1.423770235E-03  6.385243152E-03   11    28    30    0    8.268111531E-06   2.486556551E-03         1
 1      3.648996435E-02  1.104662539E+00  259   375   150  210    8.825010320E-07   5.229417259E-02        22
 2      6.327975342E-02  7.054036740E-01  202   336   150  210    1.170095614E-06   4.280255298E-02        22
 3      7.925570695E-02  1.650934236E+00  239   443   150  150    2.730759384E-07   5.229439485E-02        22
 4      1.262927260E-01  2.294190758E+00  362   540   150  150    2.943598888E-06   5.104687772E-02        22
 5      9.763108558E-02  2.885492335E+00  302   575   150  150    1.105598782E-06   5.281104869E-02        22
 6      7.535620964E-02  3.352220898E+00  214   682   150  150    1.243785196E-06   5.227487736E-02        21
 7      8.179493125E-02  2.541265342E+00  221   512   150  150    4.984719084E-08   5.227487736E-02        21
 8      8.192361208E-02  4.883308672E+00  209   741   150  150    5.736836132E-07   5.292668785E-02        21
 9      1.131692253E-01  4.410441549E+00  291   690   150  150    1.402306485E-06   5.229436307E-02        21
10      8.153524740E-02  4.787912803E+00  189   765   150  150    2.711674172E-07   5.228111333E-02        21
11      1.001592446E-01  4.010658254E+00  261   689   150  150    1.885557815E-07   5.236252535E-02        21
12      9.466107277E-02  4.694665249E+00  244   637   150  150    1.765800383E-06   5.239711233E-02        21
13      1.186249515E-01  1.733244369E+00  357   472   150  150    1.046084632E-06   5.128559319E-02        21
14      8.637992646E-02  6.151879223E+00  254   849   150  150    7.305950934E-08   5.239704884E-02        21
15      6.564088626E-02  2.488318966E+00  225   509   150  150    9.961816409E-07   5.239704885E-02        21
16      1.466496953E-01  5.856896785E+00  353   863   150  150    9.096568307E-07   5.227487736E-02        22
17      8.466206862E-02  7.009511546E+00  214   962   151  150    2.613486940E-06   5.239739808E-02        21
18      9.519842957E-02  4.947676198E+00  257   785   150  150    1.251444480E-06   5.228508058E-02        21
19      1.044507322E-01  5.704428138E+00  276   906   150  150    6.434963418E-07   5.304322660E-02        21
20      8.165461579E-02  6.106317081E+00  234   852   150  150    2.309910670E-06   5.239758858E-02        21


SIMULATION RUN TIME (SECONDS):  5.000000000E+00
TOTAL BUS COLLISIONS:      4506
TOTAL PACKETS TX:      6121

 AVG BUSBUSY             AVG USAGE              AVG IDLE
 4.610088654E-04      4.416700510E-04      3.558511247E-04

 TOT BUSBUSY             TOT USAGE              TOT IDLE
 2.821835265E+00      2.703462382E+00      2.178164735E+00


S = SIMULATED THROUGHPUT:  5.406924764E-01

G = AGGREGATE OFFEPED LOAD AS A PERCENT OF BUS CAPACITY:  1.028741091E+00

E = EFFICIENCY (S/G):  5.255865459E-01

T = THEORETICAL THROUGHPUT:  5.070834782E-01
```